Recognizing Speculative Language in Research Texts

Guillermo Moncecchi

Recognizing Speculative Language in Research Texts

Guillermo Moncecchi

Instituto de Computación
Facultad de Ingeniería
Programa de Desarrollo de las Ciencias Básicas
Universidad de la República
.
Laboratoire MoDyCo
UMR 7114 Modèles, Dynamiques, Corpus
École Doctorale Connaisance, Langage, Modélisation
Université Paris Ouest Nanterre La Défense

A thesis submitted for the degree of

Doctor en Informática - PEDECIBA
Docteur de L'Université Paris Ouest Nanterre

Jury members

Laura Alonso Alemany (reviewer), Universidad Nacional de Córdoba, Argentina
Javier Bailosian, Universidad de la República, Uruguay
Delphine Batistelli, Université Paris Sorbonne, France
Jean-Luc Minel (supervisor), Université Paris Ouest, France
Brian Roark (reviewer), Oregon Health & Science University, USA
Dina Wonsever (supervisor), Universidad de la República, Uruguay

Natural Language Processing
March 11, 2013

Jury Members

Laura Alonso Alemany (reviewer)
Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba, Argentina

Javier Baliosian
Profesor Adjunto
Instituto de Computación, Facultad de Ingeniería
Universidad de la República, Uruguay

Delphine Battistelli
Maître de conférences (HDR)
UFR ISHA, Université Paris Sorbonne, France

Jean-Luc Minel (thesis supervisor)
Professeur des Universités
Laboratoire MoDyCo UMR 7114 Modèles, Dynamiques, Corpus
École Doctorale Connaisance, Langage, Modélisation
Université Paris Ouest Nanterre La Défense, France

Brian Roark (reviewer)
Center for Spoken Language Understanding
Program in Computer Science and Engineering
Oregon Health & Science University, United States of America

Dina Wonsever (thesis supervisor)
Profesora Titular
Instituto de Computación, Facultad de Ingeniería
Universidad de la República,Uruguay

Abstract

This thesis studies the use of sequential supervised learning methods on two tasks related to the detection of hedging in scientific articles: those of hedge cue identification and hedge cue scope detection. Both tasks are addressed using a learning methodology that proposes the use of an iterative, error-based approach to improve classification performance, suggesting the incorporation of expert knowledge into the learning process through the use of *knowledge rules*.

Results are promising: for the first task, we improved baseline results by 2.5 points in terms of F-score by incorporating cue cooccurence information, while for scope detection, the incorporation of syntax information and rules for syntax scope pruning allowed us to improve classification performance from an F-score of 0.712 to a final number of 0.835. Compared with state-of-the-art methods, the results are very competitive, suggesting that the approach to improving classifiers based only on the errors commited on a held out corpus could be successfully used in other, similar tasks.

Additionaly, this thesis presents a class schema for representing sentence analysis in a unique structure, including the results of different linguistic analysis. This allows us to better manage the iterative process of classifier improvement, where different attribute sets for learning are used in each iteration. We also propose to store attributes in a relational model, instead of the traditional text-based structures, to facilitate learning data analysis and manipulation.

Para Verónica, Valentina, Manuel y Alejandro.

To Alan M. Turing.
Born June 23, 1912. Killed June 7, 1954

Acknowledgements

During this work's development, many people have helped me to grasp the linguistic aspects of speculation in scientific writing and the computational principles behind supervised classification. This thesis would have not been possible without their aid.

First of all, I would like to thank my thesis advisors, Dina Wonsever and Jean-Luc Minel for their continuous support and advice, suggesting research directions as well as invaluable methodological ideas.

The exchanges with my computer science and linguistic colleagues at the NLP group of the Universidad de la República allowed me to better understand the task I was working on and to find directions for addressing it. Especially, I want to thank Aiala Rosá for her suggestions on the incorporation of linguistic knowledge into learning, Javier Couto for helping me since, literally, the first day of my work, and Diego Garat for introducing me into the magic world of machine learning.

My thanks to Marisa Malcuori, Hugo Naya, Roser Morante, Veronica Vincze, Ken Hyland and Janet Holmes for kindly and rapidly answering my questions and requests.

I am very happy and grateful that Brian Roark and Laura Alonso have kindly accepted to review this thesis. Back in 2008, Brian taught a superb course on Sequence Analysis where I met several methods I later used for this work. Laura gave me both academic advice and outstanding encouragement when things did not seem to work.

Contents

1 Introduction **3**

1.1 Hedging in Academic Language 4
1.2 Hedge Scopes 6
1.3 The Hedging Phenomenon 8
1.4 Corpus 11
1.5 Computational Tasks for Speculation Detection 11
1.6 Objectives 13
1.7 Methodology Overview 15
1.7.1 Initial Classifiers 16
1.7.2 Improving Classification Using Expert Knowledge 17
1.8 Evaluation 19
1.9 Contributions 19
1.10 Organization 20

2 Background and Previous Work **23**

2.1 Modality and Hedging 23
2.1.1 Hedging in Scientific Texts 25
2.2 Computational Approaches to Speculation Detection 27
2.2.1 Learning Corpora 28
2.2.2 Speculation Detection as a Classification Task 31
2.2.3 Methods 33
2.2.4 Evaluation Measures 38
2.2.5 Results 39
2.3 Conclusion 43

3 Methodology **45**
3.1 Learning Scenario . . . 45
3.2 An Iterative and Error-based Learning Architecture . . . 48
3.2.1 Phase 1: Building the Training and Evaluation Sets . . . 50
3.2.2 Phase 2: Learning . . . 53
3.2.3 Phase 3: Results Analysis . . . 55
3.2.4 Phase 4: Evaluation . . . 58
3.3 Data Structures for Efficient Learning . . . 59
3.3.1 A Consolidated Structure for the Enriched Corpus . . . 59
3.3.2 Representing Instances in a Relational Model . . . 62
3.4 Summary . . . 63

4 Learning Hedge Cues and Recognizing their Scope **65**
4.1 Corpus Annotation Guidelines for Hedge Cues and Scopes . . . 65
4.2 Adding Information to the Corpus . . . 67
4.2.1 Lexical Information . . . 67
4.2.2 Hedge Information . . . 69
4.2.3 Sentence Constituents . . . 69
4.3 Training and Evaluation Corpora . . . 75
4.4 Sequential Classification Method . . . 77
4.5 Hedge Cue Identification . . . 78
4.5.1 Adding External Knowledge and Co-occurrences . . . 80
4.6 Scope Detection . . . 82
4.6.1 Baseline Classifier . . . 85
4.6.2 Iteration 1: Adding Syntax Information . . . 88
4.6.3 Iteration 2: Adding Ancestors in the Syntax Tree . . . 92
4.6.4 Iteration 3: Adjusting Ancestor Scopes . . . 96
4.6.5 Iteration 4: Handling Misclassified examples . . . 102
4.6.6 Iteration 5: Postprocessing Rules . . . 103
4.6.7 Tuning Learning Parameters . . . 104
4.7 Summary . . . 105

5 Results **109**

5.1 Performance on the Evaluation Corpus . . . 109

5.1.1 Hedge Cue Identification . . . 110

5.1.2 Scope Detection . . . 111

5.1.3 Overall System Performance . . . 113

5.2 Cross Validation . . . 113

5.3 Learning Curve . . . 116

5.4 Error Analysis . . . 117

5.4.1 Hedge Cue Identification . . . 119

5.4.2 Scope Recognition . . . 121

5.5 Evaluation on the CoNLL 2010 Shared Task Corpus . . . 124

5.6 Conclusion . . . 125

6 Conclusions **127**

6.1 Process Summary . . . 127

6.2 Methodology Remarks . . . 128

6.3 Evaluation of the Hedging Tasks . . . 129

6.3.1 Hedge Cue Identification . . . 129

6.3.2 Scope Detection . . . 130

6.3.3 Comparison with Previous Work . . . 130

6.4 Comments on the Corpus Annotation . . . 131

6.5 Future Work . . . 131

A List of Scope Classification Errors **133**

References **143**

The "unknown," said Faxe's soft voice in the forest, "the unforetold, the unproven, that is what life is based on. Ignorance is the ground of thought. Unproof is the ground of action. [...] The only thing that makes life possible is permanent, intolerable uncertainty: not knowing what comes next."

Ursula K. Le Guin

The left hand of darkness

CONTENTS

1

Introduction

> *The question and answer method seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include.*
>
> – A.Turing, *Computing Machinery and Intelligence*

A common task in Natural Language Processing (NLP) is to extract or infer factual information from textual data. In the field of natural sciences this task turns out to be of particular importance, because natural science aims to discover or describe facts from the world around us. Extracting those facts from the huge and constantly growing body of research articles in areas such as, for example, molecular biology, becomes increasingly necessary, and has been the subject of intense research in the last decade (Ananiadou et al., 2006). The fields of *information extraction* (concerning the identification of entities and their relationships from unrestricted text or speech) and *text mining* (deriving new information from those texts) have attracted particular attention, seeking to automatically populate structured databases with data extracted from text (Airola et al., 2008; Pyysalo et al., 2008; Settles, 2004). In both fields, the use of *speculative language* poses an interesting problem, because it probably reflects the subjective position of the writer towards the truth value of certain facts; when the fact is extracted or used for inference, this certainty information should not be lost. The general purpose of this work is to study and characterize the problem of speculative language within the domain of scientific research papers and to propose a methodology to detect speculative sentences and identify which fragments of the sentence are actually affected by the speculations

Consider the following sentence:

(1.1) Thus, it appears that the T-cell-specific activation of the proenkephalin promoter is mediated by NF-kappa B.

A typical information extraction system would probably extract something like:[1].

```
MEDIATE(NF-kappa B, ACTIVATION(proenkephalin promoter))
```

This seems insufficient: the system should also take into account the fact that the author presents the results with caution and avoids making any categorical assertions, and should annotate the extracted relation with some attribute to indicate that there exists some sort of uncertainty around it.

This is a general situation in scientific texts: researchers often use speculative language to convey their attitude to the truth of what is said, or as a pragmatic expression of caution. When Alan Turing, asserts something as categorical as the fact that a method can be 'suitable for introducing almost any one of the fields of human endeavour', he softens his assertion (probably seeking acceptance of the claim or anticipating readers' possible rejections) by using the term 'suggest'. The problem of the detection of speculative sentences is, therefore, a crucial one for information extraction tasks, but similar examples could easily be found in almost any other field of NLP, such as, for example, machine translation (Baker et al., 2010).

1.1 Hedging in Academic Language

Hedging, a term first introduced by Lakoff (1973) to describe the use of 'words whose job is to make things fuzzier or less fuzzy' is 'the expression of tentativeness and possibility in language use' (Hyland, 1995), and is extensively used in scientific writing. A considerable body of research has been developed about and around this linguistic phenomenon, and about the related grammatical category of epistemic modality (which makes it possible to express the degree of commitment by the speaker to what he says) (Palmer, 2001). Several authors have studied the phenomenon of speculation from philosophical, logical and linguistic points of view, collecting information from different corpora of documents in different fields, studying the surface features of hedging, enumerating common hedges, and generating a rich and heterogeneous theory about speculative language (Holmes, 1988; Hyland, 1996). All this body of work should aid in the construction of an NLP system for speculative sentence detection.

But what exactly does this detection involve? To answer this question, we need first to take a look at several examples of hedging in scientific articles in English. The first point to note

[1]Otherwise noted, every example in this thesis is extracted from the Bioscope corpus, described in following chapters of this thesis

is that more hedges are lexically marked (Hyland, 1995). The following examples show how different terms belonging to different part-of-speech classes are used to express uncertainty towards a fact.

The most widely studied form of expressing hedging in scientific research is through *modal verbs*, such as 'could', 'may', or 'should', which are epistemically used to avoid bald assertion:

(1.2) Loss of heterozygosity in PB granulocytes would be masked by the presence of significant numbers of normal granulocytes not derived from the malignant clone.

We must take into account the fact that modal verbs do not always act as epistemic devices, but also express 'root possibility', i.e. enabling conditions on a proposition (and therefore not concerning the author's attitude towards the proposition), as example 1.3 shows. This is a common behaviour: most words used as hedge cues admit other roles in the sentence, implying that it is not enough to find the term within a phrase to assert that it is expressing uncertainty.

(1.3) In one model, polymorphisms within IL-4 regulatory elements might result in overexpression of the gene, amplifying Th2 cell differentiation and class switching to IgE.

Epistemic lexical verbs such as 'suggest' or 'indicate' are often used (particularly in scientific writing) to express caution and precision when presenting a certain fact. This use introduces a characteristic of hedging we will discuss in the following chapter: hedging does not always express uncertainty, but it may also be used to express a pragmatic position.

(1.4) The findings indicate that MNDA expression is regulated by mechanisms similar to other myelomonocytic cell specific genes and genes up-regulated by interferon alpha.

Adjectives such as 'likely' (sentential) or 'putative' (attributive), adverbs such as 'probably' and even nouns such as 'hypothesis' are also used to express tentativeness:

(1.5) These results suggest that Fli-1 is likely to regulate lineage-specific genes during megakaryocytopoiesis.

(1.6) To understand the function of AML1 during B cell differentiation, we analysed regulatory regions of B cell-specific genes for potential AML1-binding sites and have identified a putative AML1-binding site in the promoter of the B cell-specific tyrosine kinase gene, blk.

(1.7) Tax-mediated transformation of T cells likely involves the deregulated expression of various cellular genes that normally regulate lymphocyte growth produced by altered activity of various endogenous host transcription factors.

(1.8) Alternative mechanisms include the possibility that NF-ATc operates on some cytoplasmic anchor or that other proteins that are controlled by calcineurin carry out the nuclear import of NF-ATc.

Hyland (1995) also mentions that about 15% of speculative sentences present what he calls *strategic hedges*: referring to experimental weaknesses, limitations of the method used or possible lack of knowledge, can be seen as a form of speculation, even when there are no explicit lexical markers in the sentence, as the following sentence shows:

(1.9) These data support the view of an impaired ligand-induced plasticity of glucocorticoid receptor regulation rather than the hypothesis of decreased glucocorticoid receptor numbers during depression.

1.2 Hedge Scopes

The identification of speculative sentences (at least those lexically marked) could be reduced to that of hedge detection: finding which words express tentativeness or possibility, and see whether they are indeed being used as speculation devices. But consider the following example:

(1.10) Since clinical remission has been observed in a significant fraction of DLCL cases, these markers may serve as critical tools for sensitive monitoring of minimal residual disease and early diagnosis of relapse.

The sentence expresses speculation, and the hedge 'may' serves to show the possibility of a certain procedure. But there is another fact within the sentence (that clinical remission have been observed in many cases) which is not hedged, so marking the whole sentence as speculative could lead to the wrong assumption that this fact is merely a possibility. The notion of *hedge scope* (Morante and Daelemans, 2009; Vincze et al., 2008) captures the idea that it is possible that only a fragment of a sentence should be included in a hedge (from now on, we will consider cases where hedging is lexically marked).

In the following example, the scope of the hedge 'may' could, for example, be the verb phrase that starts with the very word[1]:

(1.11) Since clinical remission has been observed in a significant fraction of DLCL cases, these markers {may serve as critical tools for sensitive monitoring of minimal residual disease and early diagnosis of relapse}.

In contrast with the previous topic of hedge identification, there is, to the best of our knowledge, no theoretical linguistic work on the characterization of hedge scopes (however, in the following chapter we review a general linguistic characterization of the notion of scope related with negation). The very notion of hedge scope is introduced for the Bioscope corpus annotation, and is never formally defined. Instead, a series of criteria to identify scopes based mainly on syntax is introduced. Morante and Daelemans (2009), presenting the first known system that learns scopes of hedge cues from this corpus, define the task of scope finding as: 'determining at sentence level which words in the sentences are affected by the hedge cue'. Holmes (1988) mentions a series of patterns for hedging devices that include how they are related to the propositions they modalize. These patterns do not correspond exactly to the Bioscope corpus annotation criteria (for example, they do not include the hedge as part of the scope in the case of lexical or modal verbs). In what follows, we will use the (somewhat fuzzy) hedge scope definition used for annotation of the Bioscope corpus.

To grasp how different hedges induce different sentence scopes, let us review some previous examples and introduce some new ones, where the scope of each hedge is marked.

(1.12) {Loss of heterozygosity in PB granulocytes would be masked by the presence of significant numbers of normal granulocytes not derived from the malignant clone}.

(1.13) In this review, I describe how DNA methylation and specific DNA binding proteins {may regulate transcription of the IFN-gamma gene in response to extracellular signals}.

In the first case, the scope of the hedge 'would' is the whole sentence, while in the second 'may' only affects the subordinate clause that starts with the hedge word. The difference between the two examples is a consequence of the annotation guidelines for the corpus, and

[1] In this work, hedges will be underlined and their scope marked with brackets, annotated with the name of the hedge cue in case there are multiply nested scopes

comes from the fact that, in the case of the passive voice, the subject correspond to the verb object for the corresponding active case, and so it should be within its scope (Vincze et al., 2008).

Hedge scopes can be arbitrarily nested. In fact, Hyland (1995) reported that this clustering is common in scientific writing about 43% of the hedges in his corpus occurred with another hedge in the same sentence.

(1.14) This finding $\{_{suggests}$ suggests that $\{_{may}$ the BZLF1 promoter may be regulated by the degree of squamous differentiation$\}_{may}\}_{suggests}$.

Generally, the scope of a hedge is closely related to the syntactic structure of the sentence. In example 1.14, the scope of 'suggest' is the verb phrase that includes the hedge and the clause it introduces, while the scope of the modal 'may' is the clause that includes the hedge in the syntax tree, as Figure 1.1 clearly shows.

Other examples in the corpus show that scope does not always correspond to a clause or verb phrase; in the following example, the scope of the hedge adjectives is the NP including the hedge:

(1.15) To understand the function of AML1 during B cell differentiation, we analysed regulatory regions of B cell-specific genes for {potential AML1-binding sites} and have identified a {putative AML1-binding site} in the promoter of the B cell-specific tyrosine kinase gene, blk.

In general, since hedges (as was previously shown) may belong to several different part-of-speech categories, their scopes will vary accordingly.

1.3 The Hedging Phenomenon

While hedging presents several particularities when appearing in a scientific context (those of anticipation of possible negative consequences of being proved wrong or plain politeness, beside the expression of uncertainty), the use of hedges is by no means exclusive of academic language. For example, Holmes (1988) studied how expressions used in English to express uncertainty can also be used as politeness signals in spoken and written language from a variety of contexts. The following example (taken from the previously referred article) illustrate the point:

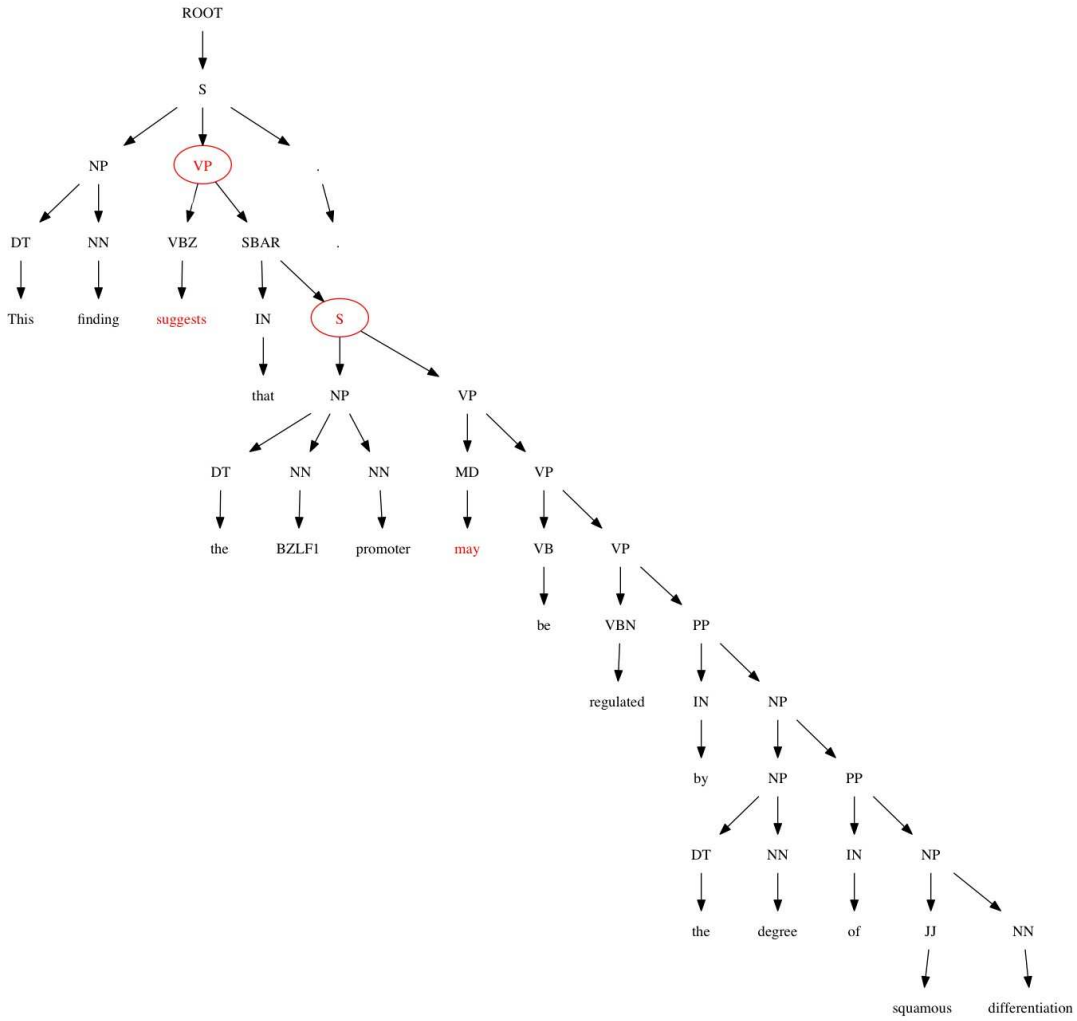

Figure 1.1: Syntax tree for the sentence from example 1.14

(1.16) I had the feeling that with all the talk about the future that perhaps some of the Ministers were not talking quite so strongly in an election year context and I wondered whether perhaps you were ...

The list of different linguistic devices identified by Holmes (1988) does not differ too much to those presented in the previous section (but it must be taken into account that the relative frequency of grammatical categories used changes, as Hyland (1995) showed).

Even when this work addresses hedging in English, we must note that expressing doubt, uncertainty and politeness is not a necessity only for English speakers (Palmer, 2001). Hedging is a cross-language phenomenon; however, the linguistic devices used to express it (and their frequency distribution) vary through different languages. While in English the use of modal verbs is the most common way of expressing uncertainty, in Spanish they are often expressed through potential mood and the subjunctive, as the following example shows:

(1.17) La delegación uruguaya, encabezada por el Vicecanciller Roberto Conde, se habría opuesto al borrador inicial que presentó Estados Unidos [1]

Liddicoat (2005) discussed the epistemic resources the French language provides to weaken knowledge claims to a greater or lesser extent, based on the study of ten research articles. Several works cited by Falahati (2004) show that hedging frequency differs across languages and cultures: for example, a study by Clyne (1991) found that German speaking authors used more hedges than English authors, no matter which language they were using.

What is probably a characteristic of hedging in academic writing, is the preeminence of hedging on propositions (or *shields*), which introduce fuzzinnes 'in the relationship between the propositional content and the speaker', in opposition to *approximators*, such as 'about 50%', which introduce fuzziness 'within the propositional content proper' (Prince et al., 1982). These led Crompton (1997) to propose to define hedges in academic writing as 'an item of language which a speaker uses to explicitly qualify his/her lack of commitment to the truth of a proposition he/she utters'. Most examples presented in section 1.1 could be seen as shields, with the probable exception of attributive adjectives, which could be seen as approximators.

[1] *The uruguayan delegation, headed by the Vice Chancellor Roberto Conde, would have opposed to the draft presented by the United States of America*, Semanario Brecha, 28/12/2012

1.4 Corpus

Before facing the computational problem of speculative language recognition, is necessary to determine the corpus to build our methods on, and to evaluate our results. While this will clearly restrict the domain and language, it seems absolutely necessary, for the two reasons presented in the previous section: hedging is expressed in different ways in different languages, and, even within the same language, the distribution of hedges is different when expressed in different domains (Holmes, 1988; Hyland, 1995).

For this work we will take the Bioscope corpus (Vincze et al., 2008), a freely available corpus of English medical free texts, biological full papers and biological scientific abstracts, as the source of empirical knowledge for hedge identification and scope detection, and as a learning source for the system developed here. The Bioscope corpus is not perfect: as previously mentioned, the definition of scope that it uses is not clear, and some annotation inconsistencies have been reported in the literature (Velldal et al., 2010). On the other hand, it has been extensively used to train and evaluate different computational approaches to the task of speculation detection, being part of the corpus for the *CoNLL 2010 Conference Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text* (Farkas et al., 2010a). For this reason, evaluating on this corpus will enable us to compare the performance of our system against state-of-the-art methods.

The corpus has every speculative sentence marked with the *hedge cues* (the linguistic devices we have called hedges so far) and their scope. As previously mentioned, all the examples in this thesis are extracted and marked exactly as in the corpus.

1.5 Computational Tasks for Speculation Detection

From a computational point of view, the task of speculation detection (at least at sentence level) can be seen as a *classification problem*: given a sentence, classify it as either speculative or not speculative. For example, the sentence

(1.18) Upon cotransfection into non-T cells, TCF-1 could transactivate through its cognate motif.

should be classified as speculative, while

(1.19) TCF-1 mRNA was expressed uniquely in T lymphocytes.

should be classified as not speculative. Most systems so far considered the presence of a hedge cue in the sentence as an indicator of speculative language (Tang et al., 2010; Velldal et al., 2010; Vlachos and Craven, 2010; Zhou et al., 2010). In the case of the Bioscope corpus, this also matches the annotation guidelines. This leads us to the first computational task to be solved:

Task 1 *Hedge cue identification: given a sentence, identify the keywords or cues in the sentence used to express uncertainty*

The aforementioned systems used a token classification approach or sequential labeling techniques, as Farkas et al. (2010b) note. In both cases, every token in the sentence is assigned a class label indicating whether or not that word is acting as a hedge cue. The following example shows how the words in example 1.14 are tagged as `H` if they correspond to hedge cues, or `O` otherwise.

(1.20) This/`O` finding/`O` suggests/`H` that/`O` the/`O` BZLF1/`O` promoter/`O` may/`H` be/`O` regulated/`O` by/`O` the/`O` degree/`O` of/`O` squamous/`O` differentiation/`O`./`O`

Additionally, it must be remembered that cues can be multi-word expressions, as shown by example 1.4. In this case, it is not enough to assign the `H` class to both words of the hedge cue: we must also model the fact that the hedge cue is the span of text viewed as a single entity. A typical approach to solve this problem is to identify the first token of the span with the class `B` and every other token in the span with `I`, keeping the `O` class for every token not included in the span. So, the class label assignments for the first tokens of example 1.4 are:

(1.21) The/`O` findings/`O` indicate/`B` that/`I` MNDA/`O` expression/`O` is/`O` ...

After labeling tokens this way, hedge cue identification can be seen as the problem of assigning the correct class to each token of an unlabeled sentence. The class of a token will probably depend not only on its own attributes, but also on features and classes of its neighbours. Intuitively, it seems more likely that the word 'that' in the previous example will be assigned the class `I` when the preceding token has been labeled as `B`. In this scenario, hedge cue identification is seen as a *sequential classification* task: we want to assign classes to an entire ordered sequence of tokens and try to maximize the probability of assigning the correct target class to every token in the sequence, considering the sequence as a whole, not just as a set of isolated tokens.

Once one or more hedge cues in a sentence had been identified, the second task to solve involves identifying the part of the sentence that they affect:

Task 2 *Scope detection: given a sentence containing a hedge cue, identify the linguistic scope of the hedge cue within the sentence*

Like hedge cues, scopes are also spans of text (typically longer than multi word hedge cues), so we can use the same reduction to a token classification task. Since scopes are longer, the usual practice (Morante and Daelemans, 2009), is to use the so-called `FOL` classes, identifying the first token of the scope as `F`, the last token as `L` and any other token in the sentence as `O`. Scope detection poses an additional challenge: hedge cues cannot be nested, but scopes (as we have already seen) usually are. Morante and Daelemans (2009) propose to generate a different learning example for each sentence hedge cue to separate the two scopes during classification. In this setting, each example becomes a pair ⟨labeled sentence, hedge cue position⟩. So, for example 1.14, the scope learning instances will be:

(1.22) ⟨This/O finding/O suggests/F that/O the/O BZLF1/O promoter/O may/O be/O regulated/O by/O the/O degree/O of/O squamous/O differentiation/L./O, 3⟩

(1.23) ⟨This/O finding/O suggests/O that/O the/F BZLF1/O promoter/O may/O be/O regulated/O by/O the/O degree/O of/Osquamous/O differentiation/L./O, 8⟩

Generating one scope for each hedge cue allows us to convert the problem of identifying nested scopes to the simpler one of recognizing two or more separate scopes. It also allows to evaluate arbitrarily nested scopes simply ‘unnestingÂ´ them first.

Learning on these instances, and using a similar approach to that used in the previous task, we should be able to identify scopes for previously unseen examples. Of course, the tasks of hedge cue identification and scope recognition are not independent: the success of the second task depends on the success of the first one.

1.6 Objectives

The general problem of classification involves classifying instances from a certain domain into one of a discrete set of possible categories (Mitchell, 1997). The function realizing this task is called a *classifier*. In probably every classification problem, two main approaches can be taken (although many variations and combinations exist in the literature): build the classifier as

a set of handcrafted rules, which, from certain attributes of instances, decide which category it belongs to, or learn the classifier from previously annotated examples, in a supervised learning approach.

The rules approach is particularly suitable when domain experts are available to write the rules, and when features directly represent linguistic information (for example, POS-tags) or other kinds of domain information. It is usually a time-consuming task, but it probably grasps the subtleties of the linguistic phenomena studied better, making it possible to take them into account when building the classifier. The supervised learning approach needs tagged data; in recent years the availability of tagged text has grown, and this kind of method has become the state-of-the-art solution for many NLP problems. In our particular problem, we have both tagged data and expert knowledge (coming from the body of work on modality and hedging, or from a human expert), so it seems reasonable to see how we can combine the two methods to achieve better classification performance.

The purpose of this thesis is to present a generalizable methodology to solve the tasks of hedge cue identification and scope detection. We have a human tagged corpus available, in which hedge cues have been identified, and their scope marked, and we want to build a statistical sequential classifier for each task. In particular, we want to investigate how we could improve classification performance, through the following methods:

- Adding useful attributes for learning; these attributes could be extracted from the training corpus or derived using external analysis tools (such as part-of-speech taggers or syntactic parsers), or semantic resources (i.e. lists of words or patterns). This is the traditional way of improving classification performance for machine learning methods.

- Exploiting previous related work: as mentioned above, there is a considerable body of linguistic work on epistemic modality and hedging; we wish to determine how that knowledge could be incorporated into the classification task. We must take into account the fact that this work does not address exactly the same phenomena as those considered here, so it should not completely overwrite the statistical classifier, but instead act as an input for learning.

- Analyzing errors: as we want to *improve* classifier performance, it seems reasonable to take a look at classification errors, and develop (with the possible aid of experts) new classification rules or incorporate new attributes. Since our decisions on improvements

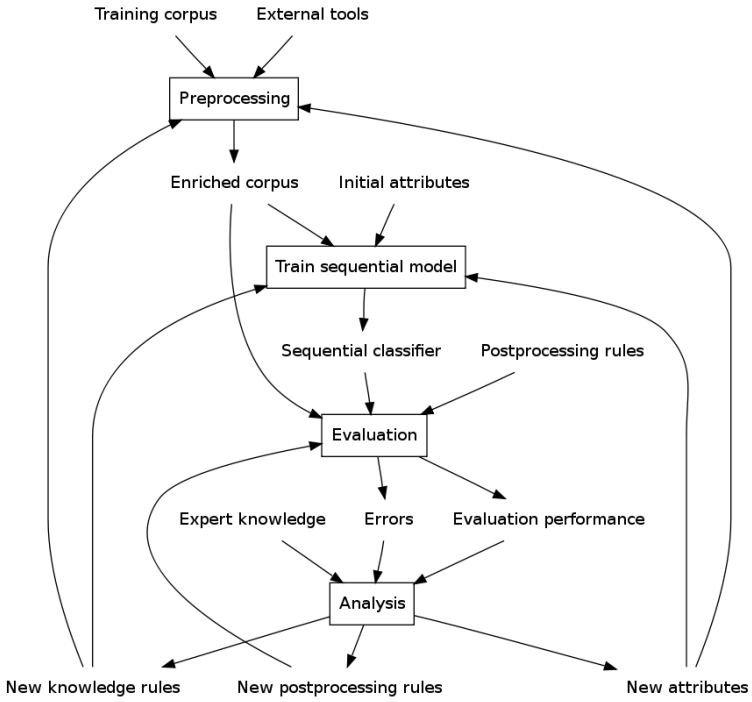

Figure 1.2: Methodology overview

will be based on classification errors, we should take into account the risk that the model we obtain will probably not be totally adequate for another corpus. To evaluate this influence, we should test our classifier on the evaluation corpus only after the final classifier has been built (i.e. do not use the evaluation corpus to obtain the classification errors to be analysed, but instead use a held out corpus sliced from the training data; this is a standard procedure for tuning learning parameters in supervised learning).

1.7 Methodology Overview

In the following sections we sketch out the proposed methodology, which is explained in detail in chapter 3. This methodology is iterative: starting from a baseline classifier, we improve classification performance by incorporating expert knowledge in the form of traditional rules or by adding new classification attributes, suggested by an expert. Figure 1.2 graphically describes the iterative process.

1.7.1 Initial Classifiers

The first step in the methodology is to convert each problem to a sequential classification one, using the methods sketched in section 1.5. Then we must select a machine learning method that can build a model from a set of labeled examples (the training set), which can be subsequently evaluated on a set of unlabeled examples (the evaluation set). There are several well studied models for supervised sequential classification, from traditional Hidden Markov Models (Rabiner, 1989) to state-of-the-art Conditional Random Fields (Lafferty et al., 2001). Our methodology can be applied using any sequential classification method (e.g. Conditional Random Fields) that can be trained on a set of features for each token. This model will usually need to be tuned, using *parameters*, whose values will be determined using previous knowledge or evaluating on a held-out corpus.

Besides a learning algorithm, we have to select which features of each token we will use to feed it. The task of *feature selection* is typical in a machine learning environment: we should select those attributes which we think, *a priori*, are the most informative with respect to the target class. In classification tasks, those features generally take values from a discrete set. To obtain the features, we can use the token itself (taking its surface form or lemma as an attribute, for example), or apply a different type of analysis to the sentence and use their results as attributes.

Hedge cues are composed of one or more words with a very clear semantic role such as, for example, verbs of modality. This suggests the inclusion of the surface form of each token, and also its lemma, as learning attributes for the hedge identification task. Its POS tag (resulting from the application of a POS tagging algorithm) and shallow parsing information should also be included, in an attempt to generalize the syntactic role of the word in the sentence. This will be the initial attributes that the learning algorithm will use to predict whether each token is part of a hedge cue, establishing baseline results. Table 1.1 shows the baseline attributes for a sentence in the hedge cue detection task.

For scope detection, besides the attributes used for linguistic marker detection, we propose to incorporate information from syntactic analysis, given that hedge cue scopes seem closely related to syntax. To do this, we must first apply a parser to the sentence instances and select, from the resulting analysis trees, some discrete features to associate to each token. Figure 1.1 depicts the syntax tree for the sentence in example 1.14, and table 1.2 shows some classification

Word	Lemma	POS	Chunk	Class
This	this	DT	B-NP	O
finding	finding	NN	I-NP	O
suggests	suggest	VBZ	B-VP	B
that	that	IN	B-SBAR	O
the	the	DT	B-NP	O
BZLF1	BZLF1	NN	I-NP	O
promoter	promoter	NN	I-NP	O
may	may	MD	B-VP	B
be	be	VB	I-VP	O
regulated	regulate	VBN	I-VP	O
by	by	IN	B-PP	O
the	the	DT	B-NP	O
degree	degree	NN	I-NP	O
of	of	IN	B-PP	O
squamous	squamous	JJ	B-NP	O
differentiation	differentiation	NN	I-NP	O
.	.	.	O	O

Table 1.1: Baseline attributes for hedge cue identification

attributes (explained in detail in Chapter 4 for one of the scope learning instances generated from the sentence.

1.7.2 Improving Classification Using Expert Knowledge

After the baseline has been set, we wish to improve classification performance by using expert knowledge. We propose to borrow methods from the active learning field (Settles, 2009) to ask experts for advice. In active learning, experts are asked to classify previous unlabeled instances, selected from a pool. In this case, we try to show the expert those instances the learner had failed to classify correctly. Unlike the general active learning case, there are no unlabeled instances, only instances wrongly classified by our current classifier. We wish to show the expert these instances and let him study them and try to discern why the present attributes fail to classify them and develop strategies to improve classification.

These strategies can be of two different kinds: the expert may suggest new attributes to feed the learning algorithm to cover the relevant linguistic phenomena, or he may suggest hand

Word	Lemma	POS	Chunk	HC	HC Start	HC text	HC Parent POS	In HC P. Scope?	Class
This	This	DT	B-NP	O	3	suggests	VP	O	O
finding	finding	NN	I-NP	O	3	suggests	VP	O	O
suggests	suggest	VBZ	B-VP	B	3	suggests	VP	F	F
that	that	IN	B-SBAR	O	3	suggests	VP	O	O
the	the	DT	B-NP	O	3	suggests	VP	O	O
BZLF1	BZLF1	NN	I-NP	O	3	suggests	VP	O	O
promoter	promoter	NN	I-NP	O	3	suggests	VP	O	O
may	may	MD	B-VP	B	3	suggests	VP	O	O
be	be	VB	I-VP	O	3	suggests	VP	O	O
regulated	regulate	VBN	I-VP	O	3	suggests	VP	O	O
by	by	IN	B-PP	O	3	suggests	VP	O	O
the	the	DT	B-NP	O	3	suggests	VP	O	O
degree	degree	NN	I-NP	O	3	suggests	VP	O	O
of	of	IN	B-PP	O	3	suggests	VP	O	O
squamous	squamous	JJ	B-NP	O	3	suggests	VP	O	O
differentiation	differentiation	NNS	I-NP	O	3	suggests	VP	L	L
.	.	.	O	O	3	suggests	VP	O	O

Table 1.2: Baseline attributes for marker scope detection - Instance 1 of example 1.14

crafted rules to solve precisely those presented instances. For example, the expert could suggest incorporating not only the POS tag of the parent of the hedge cue in the syntax tree, but also the grandparent POS tag and its scope; or he may come up with a rule that says that for the case of 'suggest', when tagged as a `VBZ` and being the first component of a `VP`, every token in the same `VP` should be included in the scope. To incorporate these rules as part of the learning process, we propose to incorporate an attribute that tags each token as `Y` if it is part of the scope suggested by the rule, and `N` otherwise (see, for example, the 'In hedge cue parent scope?' rule for the baseline classifier, in Figure 1.2).

After this analysis, the learning process starts again, retraining and reevaluating using the new attributes. The algorithm ends when no further improvement can be made, or when the expert cannot identify attributes or rules from misclassified examples.

1.8 Evaluation

The proposed approach is not risk-free: it can be prone to overfitting. Overfitting occurs when a model that fits the training examples very well, performs worse than some other model on a separate evaluation set (Mitchell, 1997). The model has learned to classify the training examples so well, that it has lost its power of generalization. An extreme example of this is a model that just memorizes the category of *every* training instance: it will be perfectly precise on the training corpus, but it will probably not generalize well on a separate corpus. Being based on error analysis, our methodology could easily overfit the training corpus if (for example) the proposed rules are too specifically tailored to each misclassified example.

To avoid this, we slice the corpus into a training and an evaluation set. During the learning process, the evaluation corpus is kept apart, and the classifier obtained in every iteration is evaluated on a held-out corpus, sliced from the training corpus, yielding new classification errors. Only when a definitive model has been built will we show performance results on the evaluation corpus, as well as how each iteration improved or reduced classification performance on it.

1.9 Contributions

We think that the key contributions of this thesis come mainly from the methodological approach we took and from the analysis of the concrete linguistic phenomenon of hedging, where

we applied it.

First, we present, apply and evaluate a methodology to improve the performance of a data-driven method using expert knowledge. The method proposes that this expert analysis is applied only to cases where the data-driven method failed in a held-out corpus. Even when nor hybrid methods nor error driven methods are new (even for the specific task we studied), the idea of explicitly stating that improvement should only come from expert analysis of errors is, to the extent of our knowledge, new.

We present a method for incorporating the obtained expert knowledge into supervised learning, through the use of knowledge rules (i.e. attributes that encode expert classification predictions for certain scenarios). The idea of including weak prediction rules for learning appears in methods such as boosting, but we include a general proposal of converting expert predictions into attributes and let the classification method determine if they are correct, based on the same learning principles used for the rest of attributes.

We also propose a method, to modify deterministically the predicted extent of a sequence given that certain parts were already identified by the classifier, and that certain conditions hold. This method, a particular type of knowledge rules, could be used in different structured learning scenarios to encode previously known relations between different parts of the structure, and we know no previous similar proposals in the literature.

We present a software architecture to combine information from different external sources and to facilitate iterative improving of classifiers, and suggest the incorporation of visualization aids to facilitate the expert work.

Once we define the methodology, we apply it to a linguistic task and study in detail how the feature engineering decisions impact on classification performance for the different proposed scenarios. The system built approaches state-of-the-art results on comparable data. We carefully analyze the results, present an error analysis of our final classifier, and make some critical comments on the corpus annotation criteria, aiming to better characterize speculation detection in Natural Language Processing.

1.10 Organization

In the following chapters we describe the tasks we wanted to address, including the linguistic and computational background, and present in detail the methodology used on them. Chapter 2 presents previous work, including linguistic studies on modality and hedging, and previous

computational approaches to the task of speculation detection. Chapter 3 presents the general methodology proposed, describing the learning scenario it applies to, its main principles and a detailed description of each of its steps. We also introduce a software architecture used to implement the methodology, aiming to better model the data structures for learning and improving time performance for the iterative process. Chapter 4 shows how we addressed the two tasks defined in section 1.5, including the corpus processing procedure, the attributes used for classification and how they were obtained, and how expert knowledge was incorporated. Chapter 5 presents the results obtained, showing how performance behaved in response to attribute and rules incorporation. Finally, Chapter 6 sums up the work, and presents final remarks, suggesting future research directions on the topic.

2

Background and Previous Work

> *For me, some of the most interesting questions are raised by the study of words whose meaning implicitly involves fuzziness –words whose job is to make things fuzzier or less fuzzy.*
>
> – G.Lakoff, *Hedges: a Study in Meaning Criteria*

In this chapter, we present the main concepts related with the detection of speculative language, especially in the scientific writing domain. We first refer to the linguistic literature on modality and its relation to the pragmatic attitude of hedging. We then present a survey of the computational approaches to speculation detection, in particular those related to hedge identification and hedge scope recognition, including the methods and features used for learning.

2.1 Modality and Hedging

It seems reasonable to study the phenomenon of speculative language in scientific writing within the logical and linguistic framework of *modality*, particularly *epistemic modality*. Modality can be defined, from a philosophical point of view, as *"a category of linguistic meaning having to do with the expression of possibility and necessity"* (von Fintel, 2006). The Cambridge Grammar of English (Huddleston and Pullum, 2002) remarks that *'The area of meaning referred to as modality is rather broad and finds expression in many areas of the language besides mood; it is, moreover, not sharply delimited or subdivided, so that we shall need to make frequent reference to the concept of prototypical features and to allow for indeterminacy at the boundaries of the categories'*.

2. BACKGROUND AND PREVIOUS WORK

Modality can be expressed using different linguistic devices: in English, for example, modal auxiliaries (such as 'could' or 'must'), adverbs ('perhaps'), adjectives ('possible'), or other lexical verbs ('suggest', 'indicate'), are all used to express the different forms of modality. Other languages, such as Spanish, can express modality through different linguistic devices, such as the subjunctive mood. Modal meaning can be further subdivided in different classes, including alethic modality (concerning *"what is possible or necessary in the widest sense"*), epistemic modality (*"what is possible or necessary given what is known and what the available evidence is"*) and deontic modality (*"what is possible, necessary, permissible, or obligatory, given a body of law or a set of modal principles or the like"*), among others.

Palmer (2001) considered modality as a grammatical category (similar to aspect or tense), which can be identified and compared across a number of different and unrelated languages. The Cambridge Grammar rather considers modality as a category of meaning, distinguishing it from the grammatical category of mood, like tense differs from time, or aspect from aspectuality. However, both sources define the semantic role of modality as concerning with the speakers' attitudes and opinions towards what he is saying, following Lyons work. Epistemic modality, in particular, applies to *"any modal system that indicates the degree of commitment by the speaker to what he says"*; this clearly includes the speaker's judgments and the warrant he had for what he said[1].

Kratzer (1981) analysed the concept of modality within the modal logic framework of *possible worlds semantics*, where a proposition is identified with the set of possible worlds where it is true. She considered that the interpretation of modals should consider a *conversational background*, which contributed the premises, and a *modal relation* which determined the 'force' of the conclusions. This implies that the meaning of modal expressions should only be determined if we take into account its background context. For the case of epistemic modality, this conversational background context would assign propositions to possible worlds, allowing to determine in which worlds the proposition holds. Morante and Sporleder (2012), citing the work of Paul Portner, mention that modal forms could be grouped into three different categories: *sentential modality*, where meaning is expressed at the sentence level, *sub-sentential modality*, where it is expressed in smaller constituents than a full clause, and *discourse modality*, where the expression of modality exceeds that of a single clause.

Sauri et al. (2006) investigated the general modality of events, which expresses the "speaker's degree of commitment to the events being referred to in a text", and defined different modal

[1]The term 'epistemic' refers here to the Greek word for 'meaning' or 'knowledge'

types, including degrees of possibility, belief, evidentially, expectation, attempting and command.

Thompson et al. (2008) identified three information dimensions concerning modality in biomedical texts: *knowledge type* indicating whether a statement is a speculation or based on evidence, *level of certainty*, indicating how certain the author is about the statement, and finally *point of view*, indicating whether the statement expresses the author's point of view or cites someone else's work or experimental findings.

Related to the concept of epistemic modality is the notion of *hedging*. The term was by Lakoff (1973), who studied the properties of words and expressions that had the ability to *"make things fuzzier or less fuzzy"*. The Concise Oxford Dictionary of Linguistics, cited by Panocová (2008), defines hedges as *"any linguistic device by which a speaker avoids being compromised by a statement that turns out to be wrong, a request that is not acceptable, and so on"*. In other words, hedges show absence of certainty, and it should be clear that they are strong indicators about the epistemic modality of any assertion.

Hedges, when lexically marked, can be considered linguistic operators, therefore inducing a scope. As we mentioned in the previous chapter, we found no linguistic study on this particular topic; however, considering the case of negation (where we can also identify negation cues), the Cambridge Grammar of English (Huddleston and Pullum, 2002), considers scope as *'the part of the meaning that is negated'*; following the same approach, we could define hedge scope as *the part of meaning in the sentences that is hedged, i.e. where the tentativeness or possibility holds*. From this perspective, scope is a *semantic* notion, that can be, however, strongly related with syntax, as the Cambridge Grammar also notes:

> Scope is in the first instance a semantic concept, and we have been identifying the scope of negation in semantic terms [...] Where a relevant component of meaning is expressed by a separate syntactic constituent, however, we can equally well refer to it in terms of its form.

2.1.1 Hedging in Scientific Texts

Since this work is mainly concerned with scientific texts, we are interested in the expression of speculation in scientific writing. In this section, we cite previous studies on the subject, seeking to show why speculation is necessary in this context and to identify the linguistic devices used to express it in the English language. This will lead us naturally to the previously defined

concept of hedging ('the expression of tentativeness and possibility in language use'). Most of this section is based on the work of Hyland (1994, 1995, 1996), who extensively studied the topic, and subsequent studies that confirmed the observation that hedging in this domain is mainly lexically marked.

Hyland (1995) observed that, in science, arguments need to be presented with caution, anticipating their possible rejection. Hedging, as the expression of tentativeness and possibility in language, becomes a valuable device for communicating unproven claims. Hedges are used to show lack of commitment to the truth value of a proposition or a desire not to express that commitment categorically.

This author suggested that the essential role of hedging in academic writing is to gain reader acceptance of claims, through three different functions:

- Expressing uncertain scientific claims with appropriate caution. Instead of writing "X produced Y", authors prefer to express "X may produce Y", actually expressing that the proposition does not correspond to proven knowledge, but derives from a plausible reasoning of the author. That is, it is true as far as is known at the moment of writing.
- Anticipating possible negative consequences of being proved wrong. Hedges, as Lakoff said, makes the relation between the author and a certain proposition fuzzier.
- Politeness: the writer is expressing that the fact he is asserting is open to discussion, appealing the readers as "intelligent colleagues", enforcing the development of a writer-reader relationship.

Hyland also showed that hedging is a frequent phenomenon on academic writing. In a corpus of scientific research articles he used for his studies, he found that hedging represent more than one word in every 50. A previous study on hedging on general conversation (Holmes, 1988) identified over 350 lexical markers of mitigation; in scientific texts the list of items was shorter, but included distinct devices such as lexical verbs (accounting for more than a half of the total instances), adjectives and adverbial forms such as 'quite', 'usually' or 'probably', modal verbs and even modal nouns such as 'possibility' or 'tendency'. The use of modal verbs in research articles is lower when compared with general writing, favouring the use of adjectives and adverbial forms. He also identified *strategic markers*, such as references to limiting experimental conditions or admission of lack of knowledge (c.f. example 1.9 in the previous chapter), accounting for about 15% of hedges in the corpus.

Subsequent work on speculation detection on academic writing confirmed those statements. Light et al. (2004) studied the use of speculative language on MEDLINE abstracts, through the expression of hypothesis, tentative conclusions, hedges and speculations. They presented several examples where sentences contained *speculative fragments*, i.e. phrases where the level of belief was less than 100%. They considered *speculative sentences* those which included one or more speculative fragments. For example, the sentence *"The level of LFB1 binding activity in adenoidcystic as well as trabecular tumours shows some variation and may either be lower or higher than in the non-tumorous tissue"* explicitly marks that the proposition there presented should be considered a possibility, from the author's perspective. After manually annotating sentences as highly speculative, low speculative and definite (proposing a set of annotation guidelines for the task), they found that about 11% of the sentences in their corpus could be considered speculative. Vincze et al. (2008) reported that, in the Bioscope biomedical corpus, about 18% of the sentences in abstracts and 19% in full articles contain speculations. This studies also confirmed the hypothesis that most speculations were realised lexically, rather than through more complex means such as the previously mentioned strategic markers.

2.2 Computational Approaches to Speculation Detection

From a computational point of view, speculative language detection is an emerging area of research, and it is only in the last five years that a relatively large body of work has been produced. Sauri et al. (2006) remarked that modality identification should be a layer of information in text analysis, to allow better inferences about events. This level of analysis seems very important when considering scientific writing: as we have previously shown, scientific assertions often include some degree of uncertainty or assessment of possibilities. Detecting epistemic modality features from identified assertions could help with concept identification and relation extraction. The expression *"Here we show that the response of the HIV-1 LTR may be governed by two independent sequences located 5' to the site of transcription initiation sequences that bind either NFAT-1 or NF kappa B"* asserts a relation between a Long Terminal Repeat and two DNA sequences. An information extraction system that omitted the analysis of modality would miss the fact that the author includes the relation under the scope of a hedge (in this case, expressed through the modal verb 'may') and that it should be presented with lower confidence. The 'either ... or' construction introduced another, different uncertainty source.

Speculative language detection had been modelled in the literature through three main tasks:

- Speculative sentence classification: given a sentence, determine its degree of certainty or speculativeness.
- Hedge cue identification: identify the linguistic devices used to express hedging within sentences. Note that solving this task implies solving the previous one, if we consider as speculative those sentences that include one or more hedge cues.
- Hedge cue scope detection: given a hedge cue in a sentence, determine its scope boundaries.

In this section, we survey the main methods used for automatic speculation detection in scientific writing, particularly in the English language (the language used in most scientific articles). First, we describe the main annotated corpora used for speculative language analysis and method evaluation, including the corpus we will use for learning. In section 2.2.2, we characterize the tree different views that allow to see speculation detection mainly as a classification problem. Section 2.2.3 presents the main computational approaches to these tasks, including rule-based and machine learning methods, and shows how linguistic knowledge had been combined with or included in supervised learning to improve performance. The last two sections of this chapter present the evaluation measures used by those methods, and their reported results.

2.2.1 Learning Corpora

In this section, we review the structure and annotations of four corpora annotated by experts with the expression of speculation. Most work on speculation detection used these corpora to build their classifiers and evaluate their results. As we will see, they represent speculation in different ways, both from a linguistic and computational point of view.

Medlock and Briscoe (2007) built a corpus of 5579 full-text papers from the functional genomics literature relating to Drosophila melanogaster (the fruit fly), annotating six papers, classifying each sentence as speculative or non speculative, for a total of 380 speculative sentences and 1157 non speculative ones. They also randomly selected 300,000 sentences from the remaining papers as training data for a weakly supervised learner. They considered a sentences as speculative when it expressed one of the following speculative phenomena:

- An assertion relating to a result that does not necessarily follow from the work presented, but could be extrapolated from it. For example, the sentence *"Pdcd4 may thus constitute a useful molecular target for cancer prevention"* (Light et al., 2004).
- Relay of hedge made in previous work (*"Dl and Ser have been proposed to act redundantly in the sensory bristle lineage*).
- Statement of knowledge paucity (*"How endocytosis of Dl leads to the activation of N remains to be elucidated"*).
- Speculative question (*"A second important question is whether the roX genes have the same, overlapping or complementing functions"*).
- Statement of speculative hypothesis (*"To test whether the reported sea urchin sequences represent a true RAG1-like match, we repeated the BLASTP search against all GenBank proteins"*).
- Anaphoric hedge reference (*"This hypothesis is supported by our finding that both pupariation rate and survival are affected by EL9"*).

The *Bioscope corpus* is a set of texts from the biomedical domain, annotated at token level for negative and speculative keywords and at the sentence level for their linguistic scope (Vincze et al., 2008). The corpus consists of three sub corpora:

- The *clinical free-texts* sub corpus consists of 1954 radiology report used for the clinical coding challenge organized by the Computational Medicine Center in Cincinnati (Pestian et al., 2007).
- The *full scientific articles* sub corpus includes the five articles of the Medlock and Briscoe corpus and four articles from the BMC Bioinformatics website.
- The *scientific abstracts* sub corpus consists of 1273 abstracts extracted from the Genia corpus.

Every document in the corpus is annotated for negations and uncertainty, along with the scope of each phenomenon. Sentence 2.1 shows the annotation of a sentence in the corpus. We can see that every hedge cue has its scope within the sentence identified and that scopes can be nested.

	Clinical	Full	Abstract
#Documents	954	9	1273
#Sentences	6383	2670	11871
%Hedge Sentences	13.4	19.4	17.7
#Hedge cues	1189	714	2769

Table 2.1: Bioscope corpus statistics about hedging

Type	Agreement
Keyword	91,46/91,71/98,05
Scope	92,46/93,07/99,42

Table 2.2: Inter-annotator agreement for the Abstracts sub corpus. The numbers denote agreement in terms of F-measure between the two student annotators, and agreements between each student and the linguistic expert

(2.1) These results $\{_{suggest}$ suggest that $\{_{likely}$ Fli-1 is likely to regulate lineage-specific genes during megakaryocytopoiesis$\}_{likely}\}_{suggest}$.

Table 2.1, extracted from Vincze et al. (2008), gives some statistics related to hedge cues and sentences for the three sub corpora.

The corpus was annotated by two independent linguists following the annotation guidelines we describe in detail in chapter 4. When the two annotations were finished, their differences were resolved by the linguistic expert who had established the annotation guidelines, yielding the gold standard labelling of hedge cues and scopes. Table 2.2 shows the inter-annotator agreement for the Abstracts sub corpus (expressed by the F-measure of one annotation, treating the second one as a gold standard) for hedge cue identification and scope recognition.

The *Genia Event corpus* (Kim et al., 2008) is a corpus of 1,000 papers, extracted from the Genia corpus (Kim et al., 2003), annotated for events at the sentence level, including 9,327 sentences and 36,114 identified events, expressing dynamic relations between biological terms (such as biological processes or regulations). Every event in this corpus is annotated for uncertainty, classifying it into three categories. If an event is under investigation or considered a hypothesis, it is marked as *doubtful*. An event is considered *probable* it its existences cannot be stated for certain. Every other event is considered *certain*.

Finally, Shatkay et al. (2008) presented a corpus where 10,000 sentences taken from full-text articles and biomedical abstracts were annotated at a fragment level, for a characterization along different dimensions, including focus (e.g. scientific versus general), polarity (positive versus negative statement), level of certainty, strength of evidence, and direction/trend (increase or decrease in certain measurement). The *certainty* dimension classified each sentence fragment into four degrees: complete uncertainty, low certainty, high likelihood, and certainty, following the annotation scheme of Wilbur et al. (2006).

2.2.2 Speculation Detection as a Classification Task

Speculation detection, from a computational point of view, include three different problem views: the first one proposes to assign each sentence a class that indicates its certainty (classifying it as speculative or non speculative, or identifying the degree of certainty it expresses); the second tries to identify the presence of hedge cues within the sentences as an indicator of speculation; the third proposes to detect the hedging scope each hedge cue induces. In this section we show how each task can be seen as a classification problem: for speculation detection, this involves assigning each sentence a class that shows its degree of certainty; hedge cue identification implies classifying each sentence token as belonging to a hedge cue while scope detection involves the identification of the first and last token of the scope.

Identifying Speculative Sentences

The first approaches to speculation detection aimed to classify each sentence as speculative or not speculative. Medlock and Briscoe (2007) proposed to use a semisupervised learning approach (that we will describe later in this chapter) to solve this binary classification task, while Shatkay et al. (2008) proposed to classify sentences with respect to their certainty dimension into the four previously mentioned degrees using supervised classifiers based on annotations.

Hedge Cue Identification

Morante and Daelemans (2009), adapting previous work on negation detection Morante et al. (2008), reduced speculation detection to hedge cue identification: a sentence would be considered speculative if it included one or more hedge cues. They modelled the task as a special case of *sequential classification*: they proposed to classify each sentence token, indicating if it was part of a hedge cue, using a `BIO` schema. The first token of a hedge cue would be assigned

class B while the remaining tokens of the cue would be assigned class I, marking every other token in the sentences with class O. The following example:

(2.2) These results indicate that in monocytic cell lineage, HIV-1 could mimic some differentiation/activation stimuli allowing nuclear NF-KB expression.

would get its tokens classified as follows (for clarity purposes, classes for tokens marked as O are not shown)

(2.3) These results indicate/B that/I in monocytic cell lineage, HIV-1 could/B mimic some differentiation/activation stimuli allowing nuclear NF-KB expression.

For the special case of noncontiguous hedge cues (such as 'either ... or') tokens other than the first one in the hedge cue were marked with a special D class, to distinguish them from those cases where a sentence included two different hedge cues (Roser Morante,personal communication). For the sentence:

(2.4) This indicates an increase in either episomal DNA or concatameric linear DNA.

the classes assigned to the sentence tokens are the following:

(2.5) This indicates/B an increase in either/B episomal DNA or/D concatameric linear DNA. }

Hedge Scope Detection

For the scope detection task, Morante and Daelemans (2009) proposed to address it in a similar way, but using FOL marking: given a sentence and a hedge cue, assign class F to the first token of the predicted scope, and L the last one, marking every other token with class O. For example 2.2, the scope detection classes would be the ones shown in 2.6 and 2.7. Note that *two* instances would be generated, one for each hedge cue present in the sentence. The classifier input included both the sentence and the hedge cue since a sentence could include two or more cues, as the previous example showed. For this task, only those sentences where a hedge cue had been found were as learning instances.

(2.6) These results indicate/F that in monocytic cell lineage, HIV-1 could mimic some differentiation/activation stimuli allowing nuclear NF-KB expression/L.

(2.7) These results indicate that in monocytic cell lineage, HIV-1 could/F mimic some differentiation/activation stimuli allowing nuclear NF-KB expression/L.

Rei and Briscoe (2010) proposed an alternative tagging schema: besides tagging the first and last tokens of the scope, they assigned class I to the tokens within the scope, in a so called FILO schema. In the previous example, sentences would be tagged as is in 2.8 and 2.9

(2.8) These results indicate/F that/I in/I monocytic/I cell/I lineage/I,/I HIV-1/I could/I mimic/I some/I differentiation/activation/I stimuli/I allowing/I nuclear/I NF-KB/I expression/L.

(2.9) These results indicate that in monocytic cell lineage, HIV-1 could/F mimic/I some/I differentiation/activation/I stimuli/I allowing/I nuclear/I NF-KB/I expression/L.

The reader could wonder why using different tagging schemas for hedge cue identification and scope detection. The answer is that, while multiword hedge cues are generally two or three tokens long, scopes are generally longer, making difficult for sequential learning methods to identify the whole span. The FOL or FILO schemas reduce the problem of span identification to that of correctly classifying their first and last tokens, or the first, interior and last tokens, respectively. This approach has been standard for tasks such as semantic role labelling or dependency parsing (Buchholz and Marsi, 2006; Surdeanu et al., 2008)

2.2.3 Methods

Once speculation detection had been characterized as a classification task, different approaches had been used to build classifiers, ranging from pure deterministic rule-based methods to machine learning methods that learn the classifier from training data, often incorporating linguistic knowledge. In this section, we survey some of these methods, also describing the features they used.

Rule-based Classification Methods

In *rule-based* approaches, the classifier takes the form of a set of hand coded rules that, based on linguistic knowledge and training data analysis, are used to classify each instance into the correct classes for the proposed task.

2. BACKGROUND AND PREVIOUS WORK

A very basic example of this kind of methods was the one used by Light et al. (2004) to develop their baseline classifier: they classified a sentence as speculative if it included one or more of the following strings: *suggest, potential, likely, may, at least, in part, possibl, potential, further investigation, unlikely, putative, insights, point toward, promise, and propose*.

A more elaborated approach was that of Kilicoglu and Bergler (2008). In this article, the authors assigned scores to sentences to indicate their uncertainty level. They took the lexical surface realizations described in Hyland (1998), composed of epistemic verbs, adjectives and nouns, and expanded it using a semi-automatic procedure, using two lexicons: the Wordnet lexicon (Miller, 1995) was used to find synonyms of epistemic terms in the core lexicon, while the UMLS SPECIALIST lexicon (McCray et al., 1994) allowed to extract nominalizations of epistemic verbs and adjectives. Every new term was added to the lexicon of surface realizations of hedges, including *unhedgers*, terms expressing strong certainty, such as the verb 'demonstrate'. After applying these procedures they produced a hedging dictionary consisting of 190 features, where each term was a 'certainty weight' ranging from 1 to 5 (where the value represented the hedging strength, assigning the highest value to prototypical hedging devices, such as epistemic verbs). The presence of one or more hedges in a sentence allowed to assign the sentence a score indicating its hedging status. They also took into account the role of syntax and developed a set of syntactic patterns that modified the hedge score of a sentence. For example, the presence of an infinitival clause together with an epistemic verb (such as, for example, 'appear to affect') increased the hedging score of a sentence, while its absence decreased it. To build these patterns they incorporated information about sentence syntactic structure. In a posterior article Kilicoglu and Bergler (2010) applied a similar method for identifying hedge cues and detecting their scopes.

Özgür and Radev (2009) used part-of-speech information for hedge cues and the syntactic structure of sentences to identify hedge scopes in the Bioscope corpus. For example, they proposed to mark the scope of a modal verb as the verb phrase in the syntax tree to which it was . Similar rules were for adjectives, determiners and conjunctions, considering also the special cases of passive voices and verbs followed by an infinitival clause.

Øvrelid et al. (2010) and Velldal et al. (2010) achieved competitive performance using a set of handcrafted syntax-based rules, based on the corpus annotation guidelines, for the task of scope detection in the CoNLL-2010 Shared Task (Farkas et al., 2010b).

Machine Learning Methods

While rule-based approaches appeared highly accurate both for hedge cue detection and scope recognition (as section 2.2.5 shows), the most successful systems in terms of the tradeoff between precision and recall had been those based on *supervised classification*. Those methods propose to build a model from learning instances of the training corpus where the intended target class had been manually identified. Methods for supervised classification are diverse, ranging from linear classifiers such as Support Vector Machines (Cristianini and Shawe-Taylor, 2000) to statistical methods such as Conditional Random Fields (Lafferty et al., 2001).

Light et al. (2004), after the manual annotation of a set of sentences as low speculative, high speculative and definite, used a SVM-based test classifier to select speculative sentences, using term based representation vectors.

Medlock and Briscoe (2007) proposed a *weakly supervised* approach for classifying sentences as speculative, working on the previously described corpus they built for the task. They started from a small set of annotated instances and, using a probabilistic model and a bigger set of unlabelled instances, inferred a set of training samples, where target classes were suggested by the model. This expanded set was, in turn, used to train a supervised classifier. The features used for learning were just single terms, based on the observation that most hedge cues took this form. Later Medlock (2008a) improved classification using stemming features and incorporating bigrams to the learning task.

Szarvas (2008) training on the same data and evaluating on newly annotated four articles on the same domain, used a Maximum Entropy classifier, incorporated bigrams and trigrams features, and selected the most informative features based on their frequency, discarding those features that did not appear as frequently as the two highest ranked candidates for each sentence.

Morante and Daelemans (2009) used a memory-based learning method for the task of hedge cue identification, using as features lexical information (lemma, word, part-of-speech) and chunking tags for each word and its context. For scope detection, the authors combined through a metalearning algorithm the results of three classifiers for hedge cue identification: a memory-based supervised inductive algorithm for learning classification tasks, a SVM-based algorithm and Conditional Random Fields. Features for learning included chain of words of the hedge cue and its neighbour tokens, chains of POS tags and binary features indicating if there were commas, colons, semicolons, verbal phrases, the presence of certain selected words

(mostly conjunctions) between the hedge cue and the token in focus and the location of the token relative to the hedge cue (pre, post, same).

As Farkas et al. (2010b) noted, the top ranked systems for the task of hedge cue identification in the CoNLL-2010 Shared Task used a sequence labelling approach (where hedge cue tokens were considered as part of a sequence, instead of trying to classify each one in isolation). Most systems used BIO classes for tagging tokens. Tang et al. (2010) presented a supervised sequential learning algorithm based on Conditional Random Fields to learn BIO classes from lexical, and shallow parsing information. They also included positional information of each token with respect to negation signals. The remaining systems used similar features for learning, some of them including information from external sources, such as Wordnet synonyms for the hedge cues in the training set or using an external dictionary (Zhou et al., 2010). The most common methods used for learning were Conditional Random Fields and Support Vector Machine based. (Li et al., 2010) presented a different approach, based on the average perceptron algorithm, using only unigrams, bigrams and trigrams of words and POS-tags while Velldal et al. (2010) used a Maximum Entropy method to build the classifier.

For scope detection, Morante et al. (2010) proposed a sequence classification approach for detecting boundaries. The used attributes included lexical information, dependency parsing information, and some features based on the information in the parse tree. They decided to include these attributes since they were used by the annotation guidelines for the Bioscope corpus used them, particularly those concerning the syntactic construction of the clause (passive voice use, subordination or coordination). Other works in this task proposed similar methods and features, generally including linguistic information in the form of postprocessing rules, as the following section shows.

Improving Learning Using Linguistic Knowledge

Many methods for NLP tasks cannot be strictly classified as rule-based or machine learning based. Instead, they often combine both approaches, incorporating some sort of rules to improve learning results. Supervised learning methods have shown to be particularly useful in many NLP tasks, but their performance generally depends on the availability of enough tagged data. Rule-based methods, on the other hand, are particularly well suited when we have domain experts available to write the rules, which is generally a time-consuming task. *Hybrid approaches* try to combine both scenarios, improving classification results aided by handcrafted rules or incorporating domain knowledge in the form of features. In this section, we show that

this is indeed what many systems did for speculation detection, particularly when addressing the scope detection task.

Trying to identify hedge cues, Tang et al. (2010) added attributes to indicate that a token was part of the pairs 'neither ... nor' or 'either... or'. Li et al. (2010) and Velldal et al. (2010) included postprocessing rules on chunks to identify multiword cues, based on the observation that this type of cues were very infrequent in the training corpus. Velldal et al. (2010) included rules to convert multiple words in just one hedge cue, for the most frequently occurring hedge cues in the training corpus.

When they presented the first attempt to automatically detect hedge cue scopes, Morante and Daelemans (2009) proposed to postprocess the classifier results, to correct those cases where it failed to predict exactly one first and one last scope token (we explain this method in detail in chapter 4). The rules they presented used the position of the hedge cue to adjust these values. For example, one of the rules stated that if no token had been predicted as `F` and more than one as `L`, the scope would start at the hedge cue and would end at the first token predicted as `L` after the hedge signal.

Since this scenario for scope detection as sequential classification is used in every work we are aware of, they usually include this postprocessing step to their learning process, including also rules to assure that scopes are continuous and do not overlap. Tang et al. (2010), for example, searched from a list of sentence end words extracted from the training corpus to guess the last token of a scope, when the classifier failed to infer it.

A very interesting hybrid method for scope detection is that proposed by Rei and Briscoe (2010): they first constructed a set of manual rules, based on annotation guidelines and using grammatical relations and part-of-speech tags. For example, the rule for `RB` pos-tag (adverbs) indicates marking as scope everything that is below the parent and after the cue. These rules were used to predict scopes. After that, the tagging sequence was fed as input to a CRF classifier, with other lexical and syntactical attributes, to produce the final classification. They also included as attributes extracted from external resources, such as a list of potential clause ending words (e.g., 'instead', 'moreover').

Li et al. (2010) combined a CRF classifier and rule-based patterns for hedge detection. The classifier was based on lexical and chunk information, and the rules where used to extend hedge cues to multiwords. For example, if the first token of a `NP` chunk tag were annotated as `B`, then the whole chunk was considered a hedge cue.

2.2.4 Evaluation Measures

To evaluate classification performance for the different tasks, several standard measures have been presented in the literature. In this section, we review how we can measure performance for the three tasks we have identified in speculative language detection.

Classification of sentences as speculative is generally carried out at the sentence level: we consider a result correct if its classification matches the correct class. The proportion of instances in the corpus correctly predicted, or *accuracy*, measures general performance of the method used. The problem with accuracy is well known: when we have a skewed distribution of positive and negative instances (in this task, about only one in five sentences is speculative), a classifier that simply predicted the majority class for every instance could achieve competitive accuracy. To overcome this, *precision* and *recall* values are generally used (Van Rijsbergen, 1979). Precision measures the proportion of correct predictions for the positive class (or True Positives), while recall indicates the proportion of correctly classified positive instances. This two measures are related: a more precise method would probably have lower recall, and vice versa. A common tradeoff measure is F-score, defined as the harmonic mean between precision and recall. This measure is the most commonly used to evaluate sentence classification. An alternative proposal is to consider several settings for classification and take the Break Even Point (where precision equals recall) as the tradeoff result.

Hedge cue identification and scope detection can be considered (as we have previously noted) a sequential classification tasks: we have to evaluate how correct is the sequence of classes predicted for the sentence tokens. In these cases, we can also use precision and recall: we consider as a True Positive the sequence of tokens corresponding to a hedge cue (or a scope) whose `BIO` or `FOL` classes have been *exactly* predicted. That is, if we predict the `B` class for the 'indicate' token and miss to predict the `I` class for the following 'that' token of the hedge cue, classification is incorrect. This is a extremely strict metric, but it has the advantage of being straightforward and unambiguous (Farkas et al., 2010b).

Note that, for the scope detection task, we have to predict a scope for each instance (since the training set only includes sentences where a hedge cue has previously been identified). In this scenario, every incorrectly predicted scope (or False Positive) necessarily involves a non predicted one (the correct scope), yielding a False Negative. This causes that, fixed the hedge cues, precision, recall (and, in consequence F-score) are the same.

Since the scope detection task depends on the previous task of hedge cue identification, we have two possible ways of evaluating its performance: using the gold-standard hedge cues (i.e. considering as attributes the hand-labelled classes of hedge cues), or using the hedge cue classes predicted by the first classifier. The first approach evaluates the ability of the classifier to do its work while the second allows us to evaluate the speculation detection system conformed by both classifiers as a whole. Both results have been reported in the literature.

2.2.5 Results

In this section, we review the results reported for the presented speculation detection methods. We will see that there is room for improving, specially for the scope detection task.

Light et al. (2004) compared a SVM classifier on sentences with a baseline system based on simple substring matching and found that precision was higher for the SVM classifier (84% compared to 55% of the baseline classifier), but recall results were clearly lower (79% of the substring classifier compared to 39% of the SVM classifier). This behaviour is consistent through the literature: the problem of data-based methods is its low recall, probably due to the lack of enough training data.

Medlock and Briscoe (2007) and Medlock (2008b), by using their weakly supervised approach to hedge detection, improved about 20 points in terms of Break Even Point values, compared with the results of the baseline algorithm of Light et al. (2004). They reported that use of POS tags did not improve classification performance in a significant way while incorporating lemma representation and bigrams produced better results. After error analysis, they found that most of them expressed knowledge paucity (such as *"The role of the roX genes and roX RNAs in this process is still unclear."*), and suggested incorporating specific knowledge paucity seeds to the weakly supervised process.

The Maximum Entropy approach of Szarvas (2008) achieved the best results on the corpus, as Table 2.3 shows. analysing errors, authors suggested that the use of more complex features such as dependency structure or clausal phrase information could help on scope detection, but probably not for the hedge identification task. They also found that the incorporation of cooccurrence information for hedge cues did not seem particularly useful.

The rule-based system of Kilicoglu and Bergler (2008) produced the same peak result. The success of this method suggests that hedging in scientific articles is expressed through simple linguistic devices, including lexical and syntactic means. analysing errors, they confirmed that difficulties in hedge cue identification come from two main sources: it is extremely difficult to

System	BEP
Baseline (Light et al., 2004)	0.60
(Medlock and Briscoe, 2007)	0.75
(Medlock, 2008b)	0.82
(Szarvas, 2008)	**0.85**
(Kilicoglu and Bergler, 2008)	**0.85**

Table 2.3: Summary of classification results on the Medlock and Briscoe (2007) corpus.

predict hedge cues when they do not appear in the training corpus, and hedge cues have word sense ambiguity.

Table 2.3 summarizes results on the corpus, expressed in terms of BEP since this was the measure used by the different authors. Szarvas (2008) also reported an F-measure of 0.85.

The development of the Bioscope corpus introduced new possibilities and challenges to the task of speculation detection. To evaluate performance, recall that the reported inter-annotator agreement for hedge cue identification on the sub corpus composed of abstracts of biological articles was 0.92 while on the full scientific articles sub corpus was 0.91. For the scope detection, the agreement values were 0.94 and 0.89, for the respective sub corpus.

Morante and Daelemans (2009) reported an F-measure of 0.69 on the abstracts section of the Bioscope corpus and 0.59 on full text articles, using their metalearning approach for the hedge cue identification task. They noted that it was extremely difficult to learn new hedge cues: almost no one of the hedge cues in the evaluation corpus that were not present in the training corpus, were correctly classified, suggesting that hedge cue identification is a highly domain dependent tasks, as Szarvas (2008) previously noted.

The SVM-based methods of Özgür and Radev (2009) achieved top performance using the all the different features presented in section 2.2.3, with a reported F-measure of 0.92 on the abstracts section of the corpus (very close to the inter-annotator agreement upper-bound), and 0.83 on the full papers section.

Both previous articles compared their results with a baseline classifier based on string matching: the one presented by Morante and Daelemans (2009) predicted as a hedge cue every word in the following list: *appear, apparent, apparently, believe, either, estimate, hypothesis, hypothesize, if, imply, likely, may, might, or, perhaps, possible, possibly, postulate, potential, potentially, presumably, probably, propose, putative, should, seem, speculate, suggest, support,*

System	F-score (abstracts)	F-score (full text articles)
Baseline1 (Morante and Daelemans, 2009)	0.71	0.64
Baseline2 (Özgür and Radev, 2009)	0.67	0.47
(Morante and Daelemans, 2009)	0.85	0.72
(Özgür and Radev, 2009)	**0.92**	**0.83**
Inter-annotator agreement (Vincze et al., 2008)	0.92	0.91

Table 2.4: Summary of hedge cue identification results, using the abstracts section and the full text articles section of the Bioscope corpus

suppose, suspect, think, uncertain, unclear, unknown, unlikely, whether, would while Özgür and Radev (2009) used the same set of strings used by Light et al. (2004). Results are summarized on Table 2.4. We can see that both systems improved with respect to the baseline classifier and were only slightly lower compared with the inter-annotator agreement for the corpus.

For the scope finding task, Morante and Daelemans (2009) reported a token F-measure of 0.89 and 0.59 on the abstracts and full text sections of the corpus while Özgür and Radev (2009) (by using a rule based approach) obtained accuracy of 0.80 and 0.61 respectively. Since results are reported using different measures, they are not comparable. The first work also reports the percentage of correct scopes: a scope is correct if all the tokens in the sentence have been assigned the correct scope class for a specific hedging signal. This measure was 0.77 in the abstract section and 0.48 on the papers section, using gold-standard hedging signals. This is about 15 points below the inter-annotator agreement for the corpus in abstracts section while results on the full text articles section are considerably lower. The most probable reason for this is that the expression of hedging and the structure of sentences is simpler and more standard in abstracts (where the author usually aims to state the most important concepts of the work) than in full text articles.

The CoNLL-2010 Shared Task on Learning to Detect Hedges and their Scope in Natural Language (Farkas et al., 2010b) allowed researchers to present their methods for uncertain sentence detection. Task 2 of the Shared Task proposed solving the problem of in-sentence hedge cue phrase identification and scope detection in two different domains (biological publications and Wikipedia articles), based on manually annotated corpora. The biological training set for the task consisted of the biological part of the Bioscope corpus, including abstracts from the GENIA corpus (Kim et al., 2003), five full articles from the functional genomics literature

Name	Precision	Recall	F-measure
(Tang et al., 2010)	0.817	**0.810**	**0.813**
(Zhou et al., 2010)	0.831	0.788	0.809
(Li et al., 2010)	0.874	0.734	0.798
(Velldal et al., 2010)	0.812	0.763	0.787
(Zhang et al., 2010)	0.821	0.753	0.785
(Ji et al., 2010)	0.787	0.762	0.774
(Morante et al., 2010)	0.788	0.747	0.767
(Kilicoglu and Bergler, 2010)	**0.865**	0.677	0.760

Table 2.5: Biological cue-level results for Task 2 of the CoNLL-2010 Shared Task

and four articles from the open access BMC Bioinformatics website, for a total of 14,541 sentences. The evaluation data set, in turn, was based on 15 biomedical articles from the PubMed databases, containing 5003 sentences, out of which 790 where uncertain. The evaluation criterion was in terms of precision, recall and F-measure, accepting a scope as correctly classified if the hedge cue and scope boundaries were both correctly identified. We will show the main results on the biological domain since this will be exactly the task on which we aim to apply the methodology we are proposing.

The best result on hedge cue identification (Tang et al., 2010) obtained an F-score of 0.813 using a supervised sequential learning algorithm, based on CRF. They tried to combine this classifier with a large margin based one, but results were actually worse. Table 2.5 summarizes the results for the best systems of the task.

For scope detection, Morante et al. (2010) obtained an F-score of 0.573, using also a sequence classification approach for detecting boundaries, based on a heuristic approximation of nearest neighbour search. The main error causes they reported concerned the differences between training data (mainly abstracts) and evaluation data (full text articles), which suggest that scope detection is not a very portable task. They also detected that the mapping from dependency parsing to hedges scopes is not straightforward, since in some cases, for example, subordinate clauses are included within the hedge scope, when they should not. Rei and Briscoe (2010) confirmed this last observation: they found that 65% of errors where produced by their rules failing to predict the correct graph components the scope included. Li et al. (2010) suggested that incorporating dependency parsing information (on top of phrase structure and

Name	Precision	Recall	F-measure
(Morante et al., 2010)	0.596	**0.552**	**0.573**
(Rei and Briscoe, 2010)	0.567	0.546	0.556
(Velldal et al., 2010)	0.567	0.540	0.553
(Kilicoglu and Bergler, 2010)	**0.625**	0.495	0.552
(Li et al., 2010)	0.574	0.479	0.522
Baseline	-	-	0.452

Table 2.6: Scope identification results for Task 2 of the CoNLL-2010 Shared Task

lexical information) actually degraded performance for the task.

Table 2.6 summarizes the results for the best systems of the task (using predicted hedge cues). They are compared with a strong baseline, proposed by Velldal et al. (2010) that always suggest as the first scope token the hedge cue, and extends the scope until the last sentence word.

Using the same corpus and evaluation criteria, Øvrelid et al. (2010), while not improving results on the task, achieved competitive performance using a set of hand-crafted syntax-based rules. They analysed system errors, and found that most of them failed to identify phrase and clause boundaries. In a recent paper, Velldal et al. (2012) reported a better F-score of 0.594 on the same corpus for scope detection using a hybrid approach that combined a set of rules on syntactic features and n-gram features of surface forms and lexical information and a machine learning system that selected subtrees in constituent structures.

2.3 Conclusion

This chapter have shown that speculation detections has been an active research area during the last years in the NLP community, including a Shared Task in a world level conference. It must be noted that, in contrast with the rich and diverse linguistic literature on hedging, most of these systems worked on a similar corpus (specially for scope detection), converting this task in a very specific one (that of hedge cue identification and scope recognition for this corpus). However, results show that the task is far from being solved: figures are still modest, even compared with similar tasks, such as semantic role labelling, which also involved semantic processing (Morante et al., 2010).

According to the literature, the most probable cause for scope detection problems is the inability of the usual lexical and syntactic attributes to correctly predict the scopes. It seems that, depending on lexical information, different scope rules apply, and that is very difficult to learn those cases only from the available training data. The use of hybrid methods on most successful systems seems to confirm this claim. For this reason, the task seems an interesting problem to evaluate an iterative approach to gradually incorporate expert knowledge. We additionally aimed to base our methods on the analysis of uncorrectly marked cases, and see how expert knowledge could be used to improve their correct detection.

In the following chapters, we present an iterative methodology for classification, which incorporates ideas from the presented methods, such as the incorporation of linguistic knowledge into supervised classification, and error analysis as a source of new attributes, and apply it to the two tasks presented in this chapter, comparing its results with those presented in this section.

3

Methodology

Bosh! Stephen said rudely. A man of genius makes no mistakes. His errors are volitional and are the portals of discovery.

– J.Joyce, *Ulysses*

In this chapter we describe the general methodology we propose to solve the computational task presented in section 1.5. We first describe the characteristics of the learning scenario in which it applies, showing the type of problem we want to solve, the learning method used to build the classifier, and the external aids we can count on. After that, we describe the general principles of the methodology, which are detailed in the following sections. At the end of the chapter we also propose a software architecture to efficiently implement the methodology (including techniques to avoid recalculating attributes and rules in each of the proposed iterations). Throughout the sections we will use as an example, the task of hedge scope detection presented in section 1.5 and described in detail in the following chapter of this thesis.

3.1 Learning Scenario

The scenario we are considering often occurs in natural language processing tasks such as, for example, part-of-speech identification or named entity recognition, where we have a learning corpus composed of sequences of elements (such as words or letters) and where we want to assign a classification value to each element, selected from a discrete set. For the part-of-speech identification example, instances are typically sentences (seen as sequences of tokens), and classification values are the POS classes defined for the task. Given these learning instances,

we want to build (using an adequate learning method) a statistical model to classify new (previously unseen) instances, based on correlations between instance attributes and target classes. To model this scenario, we make the following assumptions:

- The task we want to solve can be formulated as an NLP *classification* one: given a set X of instances (the learning corpus), we want to learn a *classifier* function, that takes values from X and assigns them one of the elements from a discrete set S (the target class). Furthermore, we will use *supervised classification*: every element of X has its target class hand-labelled, and our classification methods will use this information to build their models and evaluate their performance. For example, in a speculative sentence identification task, instances are sentences, each of them tagged as speculative or not speculative. We also assume that there exists a deterministic, computable and known function A that takes values from X and outputs a n-uple of *attribute values* taken from a previously defined discrete *attribute set*, modelling certain characteristics of the learning instances. In the previous task, for example, attributes can be lexical (words in the sentence), syntactic (elements in the syntax tree of the sentence), semantic (presence of hedges taken from a list) or any other knowledge source we draw on. The classifier function can then be seen as the composition of this function A with a learning function that, given the set of attribute values, yields the suggested classification class for every learning instance.

- While the methodology can be applied to any classification task, some methods (such as X-Rules and certain postprocessing rules) apply only when learning instances are *sequences*, i.e. ordered sets of linguistic elements: the classifier will assign each instance a sequence of target classes, corresponding to the class assigned to each element of the instance sequence. For the POS-tagging task, each instance sentence is seen as a sequence of tokens, and the target classes correspond to the suggested POS tag for each sentence token. The two tasks on which we will evaluate the methodology are examples of this kind of problem: the task of hedge scope detection, for example, can be reduced to a sequential classification problem: given a sentence and a hedge cue, we must predict for each token of the sentence a class indicating if it is part of the linguistic scope of the hedge cue.

- There is a *supervised classification method* available that, given the attribute set associated with every element in the sequence, can build a *model* from the learning instances

and their hand-labelled classification values, encoding relationships between instance attributes and classification values, enabling to infer target classes for previously unseen instances. Models such as Maximum Entropy Markov Models (McCallum et al., 2000) or Conditional Random Fields (Lafferty et al., 2001) are well studied examples of sequential supervised classification methods, which have been shown to be very effective for several sequence learning tasks, such as Named Entity recognition (Settles, 2004) or shallow parsing (Sha and Pereira, 2003).

- There is a human *expert* on the problem domain (for natural language tasks, the expert is typically a computational linguist) who is familiar with both the task characteristics and the learning method. The main role of this expert is to analyse learning instances and suggest attributes that may be relevant for learning or classification rules for certain instances, based on data analysis or applying domain knowledge or previous experience on similar tasks.

- Finally, we assume that the task to be solved is such that the expert can, by examining instance attributes and their assigned classes (possibly aided with visualization tools) characterize the instances and suppose or infer, based on domain knowledge, the causes of the correlations between attributes and target classes that led the learning method to select certain target classes for instances. To continue with the scope detection example, consider the sentence:

  (3.1) It is {suggested that danazol has an anti-estrogenic action to the monocytes through the competition and suppression of estrogen binding sites} as seen in the estrogen target organ.

  where the scope of the hedge cue is marked, and suppose the classifier suggests the following (incorrect) scope:

  (3.2) It is {suggested that danazol has an anti-estrogenic action to the monocytes through the competition and suppression of estrogen binding sites as seen in the estrogen target organ}.

  Based on instance examination, and cognizant of the attributes used by the classifier, an expert can suggest (and this is in fact a real example, as we will see in chapter 4, that

misclassification was due to the use of the scopes of the syntactic constituents of the sentences (in this case, the grandparent node of the hedge cue in the syntax tree), while the actual hedge scope excludes (probably because of annotation criteria) the final clause (beginning with 'as seen as') of the sentence. The expert can then propose to modify the definition of constituent scopes for this analysis, excluding this type of clause.

Several NLP tasks match this definition: the classification class is learned from token attributes such as surface form, lemma or similar lexical information. On the other hand, for image recognition tasks, attributes could be a list or matrix of thousands of RGB values, making it very difficult for a human being to infer the instance characteristics based solely on their values.

The usual learning methodology for this scenario proposes that the expert, studying the section of the corpus devoted to training (in a corpus-driven analysis) or exploiting previous knowledge on the task, identifies a list of potentially useful attributes for learning. Once defined, and using the supervised classification method, a classifier is built on training data (possibly adjusting learning parameters and performing feature selection using a held-out corpus). The performance of this classifier is measured on an evaluation corpus of previously unseen instances, using some of the measures described in section 2.2.4.

3.2 An Iterative and Error-based Learning Architecture

The methodology we propose for the scenario presented here is based on two main principles: an *iterative improvement* of the classifier is achieved based on its classification errors on learning data, and a *hybrid approach* is used for incorporating domain knowledge into the classifier.

Error-based learning is an integral part of several machine learning methods. The traditional perceptron algorithm (Rosenblatt, 1958) uses classification errors on the training data to adjust its linear classifier. The *gradient descent* method is used to minimize the training error of certain hypotheses, relative to training examples. All these methods utilize errors to build their models from training data. Our approach differs from those methods in that it proposes to examine errors *after* the model has been built, and use them to suggest new attributes, in an iterative fashion.

Instead of working on the whole corpus to analyse tagged examples and suggest learning attributes, we propose to start with an initial set of attributes (derived from *a priori* knowledge

on the task) and build an initial classifier using the previously mentioned standard method. Once the classifier has been built, its performance is evaluated on a held-out corpus, generating a list of classification errors (instances where the target class predicted by the classifier differs from the actual hand-tagged target class for the instance). These classification errors are studied by the expert to detect cases where classifier attributes are insufficient to predict the target class, or where the classification method (probably due to insufficient training data) fails to characterize some general phenomenon and to derive the correct instance class. This approach is grounded on the observation that it seems useless to study the phenomena where the learning method has already succeeded in classifying correctly every sample instance using the current attribute set. Other methods have been proposed that use errors on a held out corpus to induce correction rules, such as those used for the well-known Brill tagger (Brill, 1992). Our method rests on a similar idea, but differs in how it use errors: we do not want to infer correction rules (something difficult when there is not much data available), but pass them to an expert to analyse them and suggest *more general* rules incorporating previous knowledge.

To incorporate expert knowledge (in NLP tasks, this knowledge generally corresponds to linguistic expertise) into the learning task, we propose a hybrid approach that combines the ability of the supervised classification method to induce the target class from training data and the suggestions of the expert for those cases where the classifier fails. The methodology borrows ideas from *reinforcement learning* methods (Mitchell, 1997) (where we seek to optimize the actions taken by an agent, using environment rewards to the agent decisions) and active learning methods (where an oracle, e.g. a human expert, annotates previously untagged instances, incorporating these instances into the learning set)(Settles, 2009) . Two main techniques are considered: the incorporation of new attributes for learning, and the construction of what we have called *knowledge rules*, i.e. classification rules that, based on the values of certain attributes, directly suggest the target class for each instance. Since we do not know if the proposed rule applies in every instance that has the triggering attribute values, we propose to incorporate a new attribute containing the suggested classification, and let the classification method decide if this attribute, together with the remaining features, actually acts as expected, or if training data contradicts the expert suggestion.

Knowledge rules are, in our opinion, the most innovative contribution of this thesis. Most hybrid methods (such as, for example, those presented in the previous chapter) incorporate expert knowledge through deterministic rules or use rules to derive new attributes. Some data-driven metalearning approaches, such as boosting (Freund and Schapire, 1995), combine sev-

eral rule results to boost performance, but generally these rules are simple, automatically derived, not resulting from expert analysis. The key difference in the method we propose is that experts suggest the rules exactly as in a rule-based system, and they are afterwards incorporated to the learning process. This allows to combine several, probably overlapping analysis, and check their prediction ability using the training data. We also present an extension of knowledge rules, X-Rules, that allows, for sequential learning tasks, to modify prediction based on parts of the predicted structure and certain conditions of testing instances.

Figure 3.1 shows the four phases of the learning process, and how they are executed in an iterative fashion: in the first phase, the training and evaluation corpora are defined, extracting from the original data a set of attributes suitable for learning. The second phase implements the actual learning process: from the training set and the learning method, a classifier is built. Phase 3 corresponds to the analysis of classification errors and performance measures, to obtain suggested attributes and rules seeking to improve classifier performance. After this phase, the process starts again, incorporating the new suggested attributes into the corpus. These three phases are repeated until no further improvement can be obtained. Finally, the last phase involves evaluating the final classifier on a previously unseen set of instances. In the following sections we describe each phase in detail, and show how they relate each other to yield a classifier for the learning task, based on the aforementioned principles. The instantiation of the methodology for approaching the two speculation detection tasks will be addressed in the following chapter.

The method presented here differs from most other hybrid approaches, such as those reviewed in section 2.2.3 of chapter 2 mainly in the way it allows linguistic knowledge to be incorporated into learning: instead of postprocessing results using deterministic rules, it proposes to include those rules as learning attributes. The most similar approach is that of Rei and Briscoe (2010), who proposed to incorporate a set of scope prediction tags as attributes for learning. The resulting tags are similar to ours, but the two methods differ in how rules are created: in Rei and Briscoe (2010), they are based on general observation of the learning corpus, while we propose to create them based on committed errors on the held out corpus.

3.2.1 Phase 1: Building the Training and Evaluation Sets

The aim of this phase is to derive, from the original learning data (and possibly using external analysis tools), the different sets for training the classifier and evaluating its performance. Most

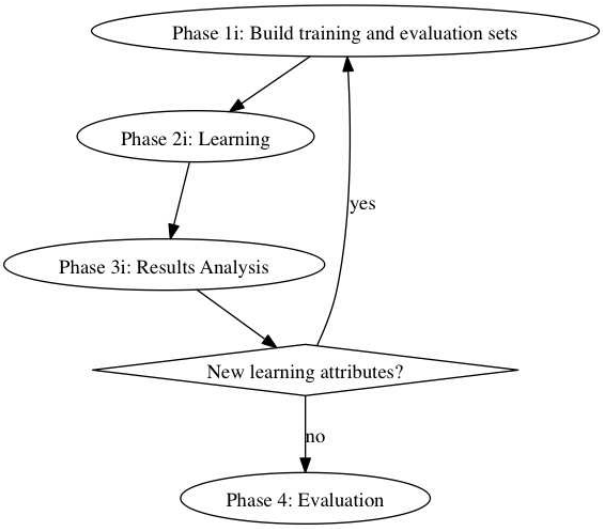

Figure 3.1: Overview of the learning process

tasks depicted in this section are typical of machine learning approaches to NLP problems, including those of learning and evaluation set creation, starting with (probably annotated) natural language text. We propose to base this creation on a unique *consolidated structure* that includes every piece of relevant information, and use it for the subsequent derivation of every attribute we want to use for learning, instead of directly generating learning attributes for the original corpus. In this way, we can solve early in the process integration problems between the various formats or structures that the information can be in.

Every piece of information we use must come from a *learning corpus*, from which we will extract every instance to be considered for learning and evaluation. Instances are, of course, different linguistic elements, depending on the task we want to solve[1]. Moreover, there is no restriction relating their format or structure, in that every attribute can be extracted from the corpus. They can be sentences, words or any linguistic data, represented with structures that best reflect the available information. Furthermore, for supervised classification we need to have each instance tagged with a certain target class, a special attribute we will use to separate instances. Like every other attribute, this target class can be explicit or may need to be extracted

[1] Actually, the methodology could be probably applied in other learning domain besides natural language. In this work we assume that tasks are Natural Language Processing tasks to fix and evaluate ideas

from the corpus. For example, in the Bioscope corpus speculative sentences are tagged in the Bioscope corpus with their hedge cues and scopes, codified in an XML nested-tags structure, as the following fragment shows.

```
<sentence id="S547.7">Because these induced gene products have
NF-kappaB sites in their promoter regions, we next examined
<xcope id="X547.7.1"><cue type="speculation" ref="X547.7.1">whether
</cue> there was an up-regulation of nuclear NF-kappaB levels
</xcope>.</sentence>
```

The information present in the learning corpus may not be enough for our learning purposes: in these cases, we need to incorporate new information, resulting from different analyses, in order to obtain a new richer corpus suitable for the task we want to solve. For example, we may want to incorporate lexical or syntactic information, such as part-of-speech tags of the words of each sentence or deep parsing trees for the learning instances. To obtain this information we may use external analysis tools, or incorporate resources related to the task (such as lists of words or expressions previously identified by studies on the domain). Since these are automatic processes, the addition of new information comes at the cost of introducing analysis errors into the corpus. This is a common problem of cascade analysis, but we expect that the presence of new information will offset the introduced errors, in terms of performance on the task we are facing. To minimize these difficulties, we must select the best tools available, incorporating, for example, those specifically trained on the domain we are working on.

Another problem with this enrichment process is that different NLP methods may use different structures for learning and use different conventions for data representation. Any mapping between those structures, classes or algorithms should be solved in this processing step. For example, a tagger and a parser may use different tokenization criteria and different sets of tags: we must realign analysis tokens and map tags to a common representation, seeking a unified view of the learning data.

After having applied the external analysis tools and consolidated the obtained information, we obtain an *enriched corpus*, which we will use for all the subsequent analyses.

Since the learning method available works on a set of learning attributes, we must provide a way of obtaining this set from the original corpus. To do this, we resort to the human expert to suggest a list of attributes suitable for the task we want to solve, and develop procedures to derive this information from original learning instances. With these attributes, we transform the corpus into a *learning set* of instances, where each original example is converted to a list of attribute/value pairs, adequate for learning using the selected method. In the special case of sequential learning, each instance is converted to a sequence of elements, each of them modelled

as a list of attribute/value pairs and tagged with a target class. For example, for the scope detection task, tokens of the sentence instances could be modelled as $\langle word, lemma, POS, chunk \rangle$ 4-uples, and have each one of them assigned a class from the $\{\texttt{F}, \texttt{O}, \texttt{L}\}$ set, expressing the membership of each token to a scope. This feature selection process concentrates every decision concerning the representation of non linear structures (such as, for example, syntax tress) to the token-per-token representation needed for learning.

This learning set is split (selecting instances in a random fashion) into three sub-corpora:

- A *training set* used for training the classifier on. All the information encoded in the model comes from this set.
- A *held-out set* used for attribute selection and parameter tuning.
- An *evaluation set* used for measuring the performance of the classifier as a solution for the task.

The portion of the learning data used for each set depends on the learning task and the amount of instances available for learning. We seek a trade-off between the size of the learning set (which should be as large as possible to help the classification method to build the best possible model) and the size of the evaluation set (whose size should be maximized to better evaluate the classifier performance). A general rule-of-thumb is to assign about three quarters of the corpus to training and the remaining quarter to evaluation. Furthermore, we must separate part of the training corpus for attribute selection and parameter tuning, reducing again the size of the training data.

3.2.2 Phase 2: Learning

In this phase (depicted in figure 3.3) we build a classifier using a list of suggested attributes on the training set. This classifier is then evaluated on the held-out corpus, obtaining the list of instances in this corpus, together with the original target class and the class predicted by the classifier. This information will be used for subsequent analysis in the next phase of the learning process.

Learning is done on the training set, on the assumption that it includes enough information to model the different phenomena found in the whole learning corpus. The role of the expert in this phase is again to suggest a list of attributes for learning (in the first iteration this list will probably coincide with the one used for building the learning set; in subsequent iterations, some attributes could be added or deleted, depending on the classifier performance, as the following section shows). The classification method will use this list to train its model on the training

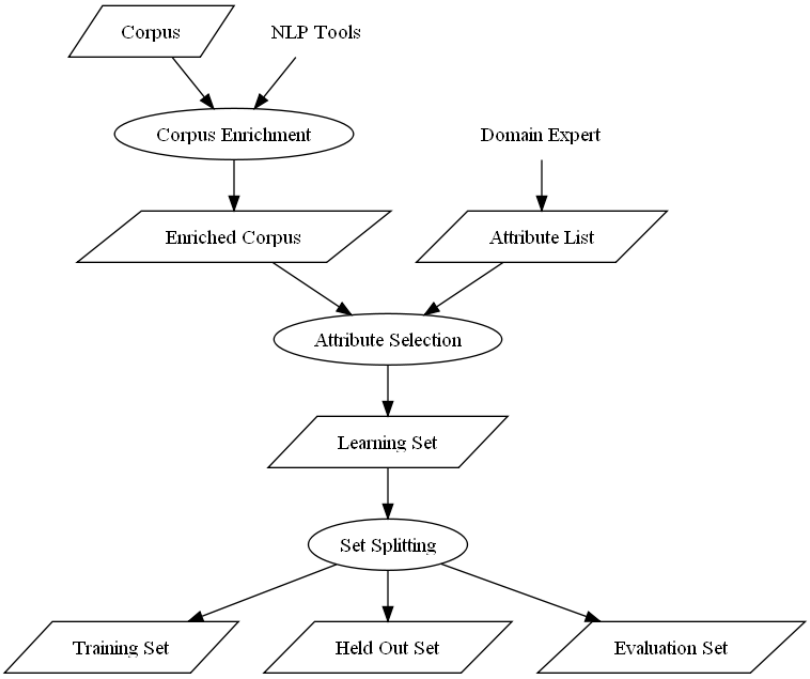

Figure 3.2: Building the Training and Evaluation Sets

set, yielding a *sequential classifier* that, given an instance represented by its attribute values, assigns it a target class value.

After building the classifier, we want to evaluate its performance on unseen data. For this purpose, we use the held-out set, comparing the classes assigned to each instance with those suggested by the classifier, and assessing performance using one of the measures presented in chapter 2, or others suitable for the selected task.

We also propose to incorporate, following Morante and Daelemans (2009), a set of *post-processing rules* into the classifier. These rules will trigger for those cases where we know that the classifier was wrong, or when we want to overwrite its classification for some special cases. For example, in a `FOL` sequence assignment, we can solve the cases where the `F` token was found but the `L` token was missed. For such cases, a postprocessing rule could assign the `L` class to the last token of the instance sequence, yielding a valid class assignment. In section 4.6, we show some cases in scope recognition where the classifier fails to identify the correct scope (probably due to insufficient training data) and we use some rules to force its correct classification. These postprocessing rules are proposed by the expert, and have the form "If

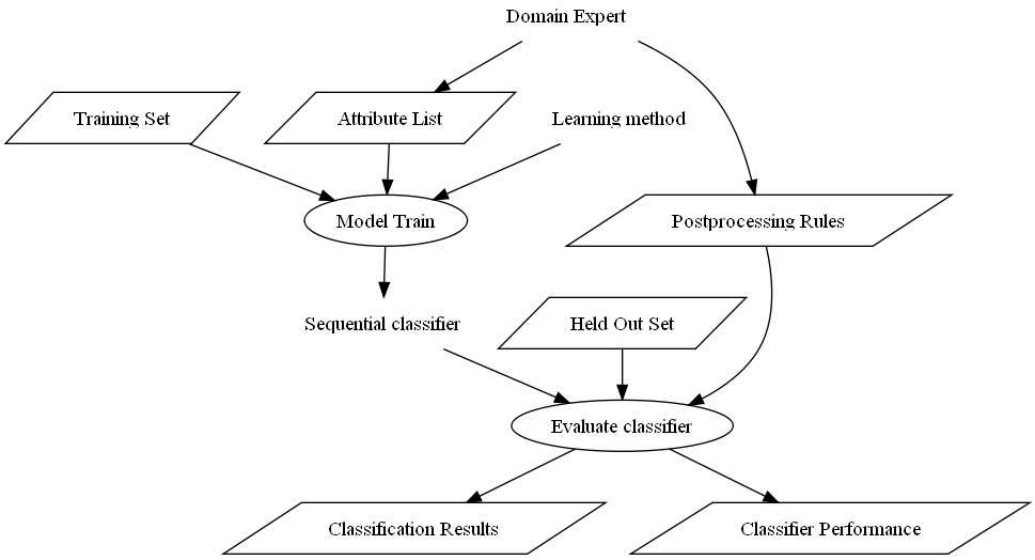

Figure 3.3: Learning

$a_1 = v_1 \ldots a_n = v_n$ then $c = C$", where $a_1 \ldots a_n$ are attributes and $v_1 \ldots v_n$ their respective values, and c is the class value assigned to the target class C.

3.2.3 Phase 3: Results Analysis

Phases 1 and 2 of the learning process are standard in a traditional classification scenario. In phase 3, a novel approach is proposed to incorporate linguistic knowledge to improve performance. The main idea is to let the expert analyse classification results on the held-out corpus,

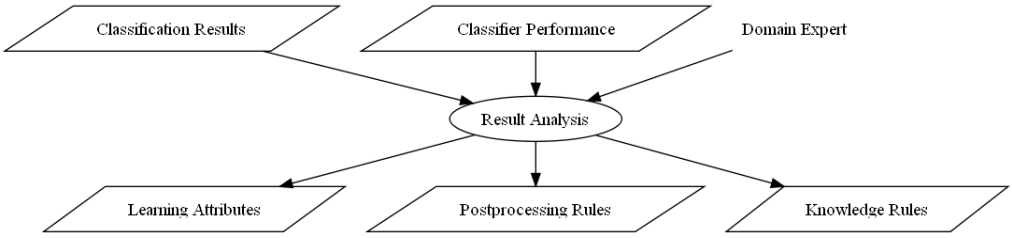

Figure 3.4: Results Analysis

and suggest new attributes and rules to better solve the task for certain, previously misclassified, instances.

Adding New Attributes

A typical form of adding new information to the learning models is through new learning features: instances are enriched with new attributes, extracted from the current set or added from previous studies on tasks similar to the one we are solving. For example, in hedge cue recognition, we may add a new attribute indicating whether the current instance (in our case, a sentence token) belongs to a list of common hedge cues. It could be objected that there is no point in incorporating attributes derivable from the current ones, but we have to take into account the fact that (if we do not have enough training data) the classification method could fail to abstract the input patterns. For example, if we consider important to identify passive voice uses for scope detection, we can add an attribute stating the presence of this type of verb conjugation in an instance sentence: this attribute can be derived from the sentence words and their POS-tags, but, as it is not a very common phenomenon, it is difficult for the model to detect its correlation with the target class. When we add the attribute, this correlation could be better identified by the sequential learning algorithm.

Knowledge Rules

Linguistic or domain knowledge can also naturally be stated as *rules* that suggest the class or list of classes that should be assigned to instances, based on certain conditions on features.

Definition 1 *Given an instance X, represented as a set of feature-value pairs F, and a boolean condition function C, that, given an instance returns $True$ if the condition holds, and $False$ otherwise, classify instance X with class Y if the condition holds on X*

For example, based on corpus annotation guidelines, a rule could state that the scope of a verb hedge cue should be the verb phrase that includes the cue, as in the expression

(3.3) This finding {suggests that the BZLF1 promoter may be regulated by the degree of squamous differentiation}.

In the previous example, assuming a `FOL` format for scope identification, the token 'suggest' should be assigned class `F` and the token 'differentiation' should be assigned class `L`, assigning class `O` to every other token in the sentence.

Since we do not know in fact if certain proposed rules always apply, we do not want to directly modify the classification results, but rather to incorporate the rule predictions as attributes for the learning task. To do this, we propose to use a similar approach to Rosá (2011),

i.e. to incorporate these rules as a new attribute, valued with the class predictions of the rule, trying to 'help' the classifier to detect those cases where the rule should fire, without ignoring the remaining attributes. In the previous example, this attribute would be (when the rule condition holds) valued `F` or `L` if the token corresponds to the first or last word of the enclosing verb phrase, respectively. We have also called these attributes *knowledge rules*, to reflect the fact that, despite being normal attributes, they encode the suggestion of a classification result based on domain knowledge.

This configuration allows us to incorporate heuristic rules without caring too much about their real prediction ability: we expect the classification method to do this for us, detecting correlations between the rule result (and the rest of the attributes) and the predicted class. To achieve this, we must add, besides the knowledge rule attribute, the necessary features to check if the C condition holds (in the previous example, we add the parent constituent tag in the syntax tree for the hedge cue, to let the classification method distinguish between verb phrases and every other possible parent constituent).

There are some cases where we do actually *want* to overwrite classifier results: sometimes we are sure the classifier has committed an error, because the results are not well-formed. In those cases, as we previously saw, we propose to add one or more *postprocessing rules* to overwrite classification results. For the scope detection example, we could include rules to assign the scope of the enclosing clause to verb hedge cues when the classifier has not exactly found one `F` token and one `L` token, as we know for sure that something has gone wrong.

X-Rules

For sequential classification tasks, there is an additional issue: sometimes the knowledge rule indicates the beginning of the sequence, and its end can be determined using the remaining attributes. For example, suppose the classifier suggests the class `scope` in the learning instance shown in table 3.1 (using as attributes the scopes of the parent and grandparent constituents for the hedge cue in the syntax tree). If we could associate the `F` class suggested by the classifier with the grandparent scope rule, we would not be concerned about the prediction for the last token, because we would know that it would always correspond to the last token of the grandparent clause. To achieve this, we modify the class we want to learn, introducing a new class, say X, instead of `F`, to indicate that, in those cases, the `L` token must not be learned, but calculated in the postprocessing step, in terms of other attribute values (in this example, using the hedge cue grandparent constituent limits). This change also affects the classes of training data instances (in the example, every training instance where the scope coincides with the grandparent scope attribute will have its `F`-classified token class changed to a new X class). We have named these special knowledge rules *X-Rules*

Hedge	PPOS	GPPOS	Lemma	PScope	GPScope	Scope
O	VP	S	This	O	O	O
O	VP	S	finding	O	O	O
O	VP	S	suggest	O	O	O
O	VP	S	that	O	O	O
O	VP	S	the	O	F	F
O	VP	S	BZLF1	O	O	O
O	VP	S	promoter	O	O	O
B	VP	S	may	F	O	O
O	VP	S	be	O	O	O
O	VP	S	regulate	O	O	O
O	VP	S	by	O	O	O
O	VP	S	the	O	O	O
O	VP	S	degree	O	O	O
O	VP	S	of	O	O	O
O	VP	S	squamous	O	O	O
O	VP	S	differentiation	L	L	O
O	VP	S	.	O	O	O

Table 3.1: Evaluation instance where the scope ending could not be identified

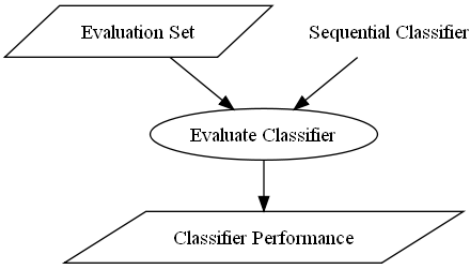

Figure 3.5: Evaluation

After adding the new attributes and changing the relevant class values in the training set, the process starts over again from Phase 1. If performance on the held-out corpus improves, these attributes are added to the best configuration so far, and used as the starting point for a new analysis. When the expert fails to suggest new rules or attributes, the process ends, yielding the best classifier so far as a result.

3.2.4 Phase 4: Evaluation

In the final phase of the process, we apply the final classifier to the evaluation set, and (in exactly the same way as in phase 2 on the held-out corpus), assess its performance using precision, recall or any other performance measure. Figure 3.5 depicts this process.

3.3 Data Structures for Efficient Learning

The learning process we propose introduces some challenges when we come to the task of efficiently implementing it: as it is an iterative process, there are several data transformation steps we must take in every iteration. Building the enriched corpus involves taking the original corpus and processing it with external analysis tools to add relevant information; the feature selection step extracts relevant attributes from the enriched corpus to feed the classification method; finally, each learning iteration introduces a new different attribute set, including the previously used ones and adding new features, suggested by the expert and extracted from the corpus. If we want to minimize the total computation time, we must provide some way of executing these steps only when needed, and avoid recalculating attributes already used in previous iterations. We present two implementation ideas to achieve this: maintaining an efficiently accessible data structure to hold all the corpus information, and keeping learning information in a relational data structure, where we have operations available to easily and efficiently recover, update and add new attributes.

3.3.1 A Consolidated Structure for the Enriched Corpus

It makes no sense to rebuild the enriched corpus every time we want to add some attribute to the learning set, since the corpus, in those cases, does not change. To cope with this situation, we propose to use some data representation techniques to allow the incremental building of the data used for representing all the information we want to work on. The original corpus usually takes the form of an annotated text, where some linguistic phenomenon is identified. If we take, for example, the corpus for hedge detection presented in chapter 1, it is originally represented as an XML structure, where documents and sentences are elements of the structure, each of them including in turn sub-elements including hedge cues and their scopes. If we want to add part-of-speech tags and an explicit representation of the deep syntactic structure of each sentences, we must somehow consolidate this information with the original corpus data.

To build the enriched corpus, we propose to build a single data structure. This consolidated structure will be the one used for attribute selection in every iteration, thus avoiding regenerating it when the iteration does not involve the incorporation of new analysis results. It will also be the input for visualization tools used for corpus analysis. In this section we present the structure, showing how to cope with the different linguistic structures commonly used in NLP tasks.

When analysing a natural language sentence, the results of linguistic analysis can be classified into the following categories:

- *Sentence-related information* is generally expressed through attributes associated with

the sentence as a whole. The speculative status of a sentence, or the indication of its use of passive voice are examples of such attributes.

- *Sentence-fragment information* may be seen as attributes related to sentence spans. For example, linguistic scopes or chunks generally comprise more than one of the sentence words.

- *Token related information* takes the form of attributes naturally associated with each sentence token. Surface forms, lemmas and POS-tags are examples of these attributes.

- *Sentence structure information* is the result of parsing analysis, yielding structures such as parse trees or graphs, as in the case of syntax constituent or dependency parsing analysis

We propose a data structure to consolidate this information, based on the sentence structure information, and including the remaining information as attributes of the nodes in the corresponding graph or tree. For this work, we assume a tree-shaped analysis structure; the extension to graphs is straightforward.

The initial structure is the tree resulting from parsing analysis. It includes a leaf for every token in the sentence, and inner nodes to represent the different sentence constituents and their relations, in the standard approach for representing syntax trees. Figure 1.1, in chapter 1 presents an example of such a structure, including POS-tags and constituents for a natural language sentence.

To incorporate token related information, we simply add an attribute to each leaf node for each feature we want to represent. The integration of these data into the structure involves solving the previously mentioned problem of different tokenization schemas.

The information related to the whole sentence or to sentence fragments must be converted to token attributes to allow its incorporation into the structure. In the case of sentence-related information, we add an attribute to each token, with the same value for all the tokens of the sentence. In the passive voice example, the attribute could be valued `Y` if the main verb in the sentence is conjugated in the passive voice, and `N` otherwise. For sentence fragments, the solution is more subtle: we must represent the information using a sequential labelling schema, such as the ones presented in the previous chapter, to indicate the initial and final tokens of the span, and the tokens in-between.

The task of scope detection introduces the additional issue of multiple span representation: sometimes, we have to represent in the structure the fact that there are several spans within a sentence, representing different occurrences of the same phenomenon. To represent this, we

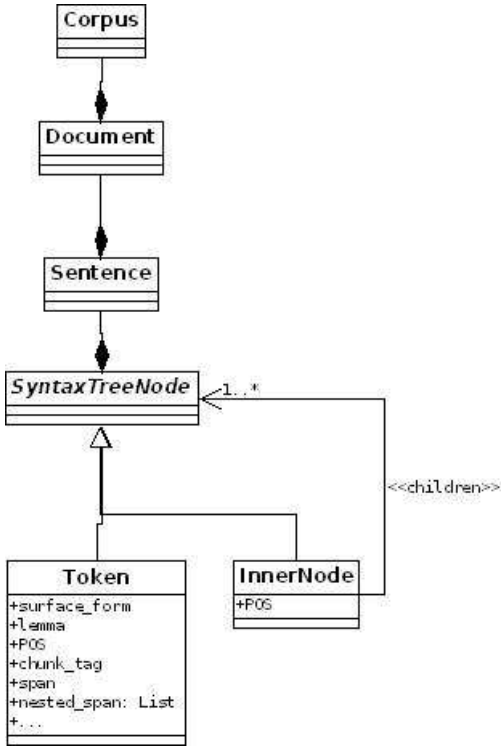

Figure 3.6: Class schema for representing sentences in a natural language corpus

propose to use list attributes, where each element of each list represents the corresponding span, using the same sequential labelling schema presented in the previous paragraph.

The class schema in Figure 3.6 depicts the main classes involved in a typical scenario, and how they are related. Note that token attributes are just typical examples of information in NLP tasks. Any token related attribute should finally be included in the $Token$ class, while the tree structure is represented through the $SyntaxTreeNode$, $InnerNode$ and $Token$ relations.

In the following chapters we present several examples of these structures, including the representation of the different linguistic phenomena we intend to use for learning. We show how we used different schemas to represent hedge cue and scope information, and present examples of different attributes used to represent the information, following the previous guidelines.

3.3.2 Representing Instances in a Relational Model

In natural language classification tasks, input data are represented as a set (or a sequence, in the case of sequential classification) of token attributes. Every learning instance is an n-uple of attributes, representing the available information about the token, one of them being the class we want to predict. In the hedge cue identification task, for example, we could take as learning attributes the surface form, lemma and part-of-speech tag for each token, and try to predict its hedge-cue tag. Standard representations of these sets are text-based structures, with a line for each token including a tab-separated list of attribute values. Examples of these structures are attribute-relation file formats (ARFF) (Witten and Frank, 2005), or CoNLL-X Shared Tasks data file formats (Buchholz and Marsi, 2006; Surdeanu et al., 2008).

While these formats are easily managed by the implementation of different learning algorithms, they are not well suited for an efficient implementation of methods such as the one we are proposing: incorporating or eliminating attributes essentially involves recreating the whole structure from the original data. We propose instead to use a different structure, based on a relational model, where data are represented in terms of tables of tuples, grouped into relations, and where there is a declarative method available for specifying tuples and querying them (Codd, 1970). With this representation, each time a new attribute for learning is added, we just have to calculate it and add it to the corresponding table, without having to recalculate the remaining attributes (whose values for each instance we already calculated in previous iterations).

This approach presents several advantages:

- Statistics can be easily computed on the learning set, through declarative statements, such as SQL queries. If, for example, we want to know the most common misclassified hedge cues, we simply group instances in the held out set, counting how many times each of them had a class predicted that was different to that assigned by the corpus annotators.
- Instances with similar values can be grouped together, facilitating the identification of common patterns in data, information that is potentially useful for the human expert analysis.
- Different attribute sets can be selected for learning, in a much more efficient way than by recomputing every attribute from the consolidated structure.
- Attributes which depend only on preexisting attribute values can be added, without needing to recompute them.
- The predicted classes can be updated after each classifier run, enabling classifier performance to be easily measured.

Of course, the learning method assumes a different data representation as an input. Since we have all the information in the relational database, we only have to generate this structure from the corresponding tables. Even if the learning method changes, we just have to change this generation process, allowing us to separate the learning data from the particular representation needed by the method. Exactly the same approach will be used for evaluation: we will generate evaluation instances by extracting information from the database and converting it to an adequate representation for the constructed classifier.

3.4 Summary

Since the problems of hedge cue identification and scope detection could be well suited to an hybrid approach (as the previous chapter showed), we aimed to develop a methodology (general enough to be further applied to other similar problems) that allowed to examine prediction problems by a human expert and incorporate the results of this analysis to improve classification performance.

We proposed in this chapter a learning architecture some feature engineering techniques that allow to incorporate expert knowledge to improve any classifier on certain scenarios. We also suggested some implementation techniques to improve time efficiency of the learning process.

In the following chapter we present the application of the methodology to the defined tasks and a detailed analysis of its success, both in terms of factibility and classifier performance.

4

Learning Hedge Cues and Recognizing their Scope

In this chapter, we show how we applied the methodology presented in the previous chapter to the learning tasks of hedge cue identification and scope recognition, presented in chapter 1. We first further describe the corpus used, including its annotation guidelines. We then show how we enriched the corpus adding lexical and syntactic information and provide a detailed description of each improving iteration until we reached the top performance classifiers for both tasks. We leave for the following chapter showing and analysing results on a previously unseen corpus.

The computational approach to both tasks was the same: start from an initial sequential learning classifier based on a base set of corpus attributes and iteratively improve it incorporating new attributes and/or postprocessing rules resulting from the analysis of classification errors or from previous linguistic knowledge.

4.1 Corpus Annotation Guidelines for Hedge Cues and Scopes

Training and evaluation data were extracted, as it was mentioned in Chapter 1, from the Bioscope corpus. In particular, we decided to use the biological abstracts sub corpus. To better understand the corpus, and to use this information as an additional knowledge source for addressing the learning tasks, we review the annotation guidelines. The following paragraphs summarize these guidelines; for a detailed explanation, the reader is referred to Vincze et al. (2008).

The criterion used for speculative sentence identification in the annotation guidelines is that it 'states the possible existence of a thing'. This criterion does not correspond exactly to

what had previously been called 'hedges' in the linguistic literature as we showed in chapter 2. For example, sentences including coordinating conjunctions such as 'or' are considered speculative for annotation, something new (to the best of our knowledge) for hedging studies. There is another implicit rule: speculation must be expressed lexically through one or more words (called hedge cues). Every example in the corpus (such as those included in this thesis) includes those hedge cues. This definition must necessarily be incomplete: sentences in the Bioscope corpus have hedge cues identified and their scope defined, but that does not mean they cover every possible source of hedging in those sentences: recall from chapter 2 that some hedges are not lexically marked. One could also argue that hedging does not always imply uncertainty, but instead an author's pragmatic attitude seeking acceptance of his claims. Being said, we anyway took this as the definition for our computational tasks, and we will use the term 'hedging' as a synonymy of 'speculation' for the rest of this thesis.

For hedge cue annotation, the guidelines established a minimalist strategy: keywords included the minimal lexical unit that expressed hedging. This means that we must include in a hedge cue enough tokens to denote speculations, but no more. For example, 'indicate that' is a multi-token expression that by itself denotes speculation, and that cannot be further divided without potentially losing its condition. Instead, 'suggest that' is not a hedge cue, since the word 'suggest' by itself, in the right context, denotes uncertainty, and, therefore, should be marked as a one-word hedge cue.

Every word having speculative content is marked as a hedge cue. Verbs such as 'appear', modal verbs such as 'may', adjectives such as 'putative', adverbs such as 'probably' and even nouns such as 'hypothesis' or coordinating conjunctions such as 'or', appear as hedge cues in the corpus if they are as speculative devices. Some complex multi word hedges even include non consecutive tokens, as the following example shows.

(4.1) In patients suffering from Cushing's syndrome, {the circadian rhythm of plasma cortisol either disappeared or was inverted} while that of GR did not significantly deviate from the normal subjects.

Table 4.1 shows the ten most common hedge cues in the corpus. We can see that it includes almost every different part-of-speech we had previously mentioned.

Every hedge cue in the corpus induced a scope, i.e. the part of the sentence it affected. These scopes were annotated using a maximal length criterion: the largest continuous syntactic unit including the hedge cue was marked as its scope. For example, in the sentence

(4.2) The data $\{_{indicate\_that}$ indicate that decrease of ER levels in cell $\{_{may}$ may involve in the pathogenesis of climacteric syndrome$\}_{may}\}_{indicate_that}$.

Hedge cue	occurrences
may	516
suggest	326
suggesting	149
indicate that	148
or	120
whether	96
appears	84
suggested	71
might	70
could	67

Table 4.1: Ten most common hedge cues in the BIOSCOPE corpus

both scopes correspond to the verb phrases headed by their respective hedge cues. Depending on the part-of-speech tags of the hedge cues, different syntactic units were used as scopes. Table 4.2 describes the list of rules for scopes specified in the annotation guidelines.

Often, scopes were nested, i.e. one of them was included within the other. However, it was not possible to have overlapping scopes: when this situation arose one of the scopes was extended to fully include the other. The following example includes a hedge cue ('can') whose scope is included within the scope of another ('indicate that'):

(4.3) Cotransfection studies with this cDNA $\{_{indicate\_that}$ indicate that it $\{_{can}$ can repress basal promoter activity$\}_{can}\}_{indicate_that}$.

4.2 Adding Information to the Corpus

Before addressing the learning task, we aimed at enriching the texts in the Bioscope corpus with results from different analyses in order to obtain a new richer corpus, suitable for use on speculative language detection tasks. To represent this information, we followed the guidelines described in section 3.3 to accommodate distinctly structured results. We started with the corpus original sentences, tokenised them and added lexical and syntactic information.

4.2.1 Lexical Information

To incorporate lexical information, each Bioscope sentence was analysed with the GENIA tagger (Tsuruoka et al., 2005), a widely used part-of-speech tagger, trained on the biological

Rule	Example
The scope of verbal elements (i.e. verbs and auxiliaries) coincides with the verb phrase that encloses them	These data {suggest that IL-4 promoter activity is normally down-regulated by an NRE via repression of the enhancer positive regulatory element}.
When verbs are used in passive voice, or in the case of raising verbs, their scope changes to the enclosing clause	The nature of the nuclear factor(s) that control TNF-alpha gene transcription in humans remains obscure, although {NF-kappaB has been suggested}. Each binding site contributes to the overall activity of the enhancer, however {no single element seems absolutely required for activity}.
The scope of attributive adjectives extends to the following noun phrase whereas the scope of predicative adjectives includes the whole sentence	Modulation of normal erythroid differentiation by the endogenous thyroid hormone and retinoic acid receptors: a {possible target for v-erbA oncogene action}. It is {possible that the CRE site is responsible for induction of bcl-2 expression in other cell types, particularly those in which protein kinase C is involved}.
Sentential adverbs scope over the entire clause, while other adverbs scope extends to the next noun phrase.	This silencer is inactive in the most immature DN thymocytes, which {probably use a distinct silencer mechanism to down-regulate CD4 gene expression}. Thus, the novel enhancer element identified in this study is {probably a target site for both positive and negative factors}.
The scope of conjunctions extends to every member of the coordination	Nucleotide sequence and PCR analyses demonstrated the presence of {novel duplications or deletions involving the NF-kappa B motif}.
In the case of hedge cues including nonconsecutive tokens, the scope is the maximal syntactic unit including every token (i.e., no scope in the corpus excludes the hedge cue tokens it is induced by)	Only the activities for NF-AT and AP-1 sites require two signals for optimal induction, i.e., PMA plus {either lectin or antibody to the CD3 or CD28 surface molecules}.

Table 4.2: Scope annotation guidelines for the Bioscope Corpus

domain. This tagger was also used to annotate named entities and chunking information at a token level. The attributes we added to each token were the following:

- Surface: the word form or punctuation symbol.
- Lemma: lemma of the surface form
- POS tag: the Penn Treebank part-of-speech tag, as described in Santorini (1990).
- Chunk tag: the IOB tags produced by the Genia tagger. IOB tags identify a sequence of tokens as an entity marking the first token with an `B` tag, the rest of tokens of the hedge cue with a `I` tag, and the rest of the sentence tokens with a `O` tag.
- NER tag: the IOB tags for the identification of biological entities within the sentence.

4.2.2 Hedge Information

Hedge cue and scope information (already present in the corpus) cannot be directly represented at a token level: it has an arborescent structure, with potentially nested scopes. We again used an IOB tagging schema for marking hedge cues and scopes. To cope with nested scopes, we associated to each token a list of tags, with the first element for the outside scope, the second element for the first nested scope, and so on. The list length is the maximum nesting level for the sentence, allowing to include arbitrary nested scopes (the maximum nesting level we measured in the corpus was two). This notation also allows to represent more complicated cases such as those where two different scopes are nested within another one. The scope attribute for the tokens in sentence 1.14 were the following:

(4.4) This/`[O,O]` finding/`[O,O]` suggests/`[B,O]` that/`[I,O]` the/`[I,B]` BZLF1/`[I,I]` promoter/`[I,I]` may/`[I,I]` be/`[I,I]` regulated/`[I,I]` by/`[I,I]` the/`[I,I]` degree/`[I,I]` of/`[I,I]` squamous/`[I,I]` differentiation/`[I,I]` ./`[O,O]`

4.2.3 Sentence Constituents

We also analysed the corpus searching for sentence constituents, using the Stanford Parser (Klein and Manning, 2003). We built a syntactic analysis tree for each sentence of the corpus. Since this parser was trained on a different domain from ours, we tried to improve its performance using as inputs for the parser the resulting tokens from the GENIA tagger analysis, and their part-of-speech tags. As the usual representation of token-per-token features did not satisfactorily accommodate the parsing information (which is essentially tree-shaped), we

#Documents	1269
#Sentences	11238
Speculative sentences	18.5%
#Hedge cues	2664

Table 4.3: Working corpus statistics after consolidating information

decided to start with the tree resulting from the syntactic analysis, decorating each of its leaves (containing sentence tokens) with the remaining features.

The main issue in synchronizing the three sources of information was tokenization and tag set selection: if we could not manage to tokenise the sentences in exactly the same way by the different analyses, integrating them into one structure would not be possible. Fortunately, the tagger and parser used the same tag set and conventions for tokenization (those used for annotating the PennTreebank), so we followed the same approach when tokenising the Bioscope sentences. Even then, certain problems arose with GENIA incorrectly tokenising some sentences, or not following exactly the tokenization conventions. GENIA results were post-processed, using ad-hoc rules, to correct these situations, but for about 5% of the sentences (20% of them containing hedges) we could not align both tokenizations, so we decided to eliminate them from the working corpus. Table 4.3 shows the final number of documents, sentences and hedge cues we worked on.

The class diagram in Figure 4.1 depicts the data structure created for holding the information, where only the main attributes are shown. Figure 4.2 shows the analysis tree decorated with the part-of-speech, chunk, NER and hedging features for sentence 1.14.

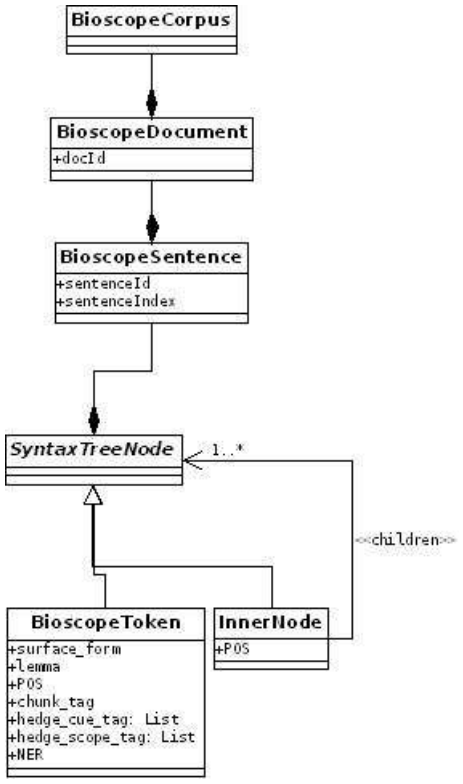

Figure 4.1: Class diagram for the data structure used for representing the corpus and its analysis results

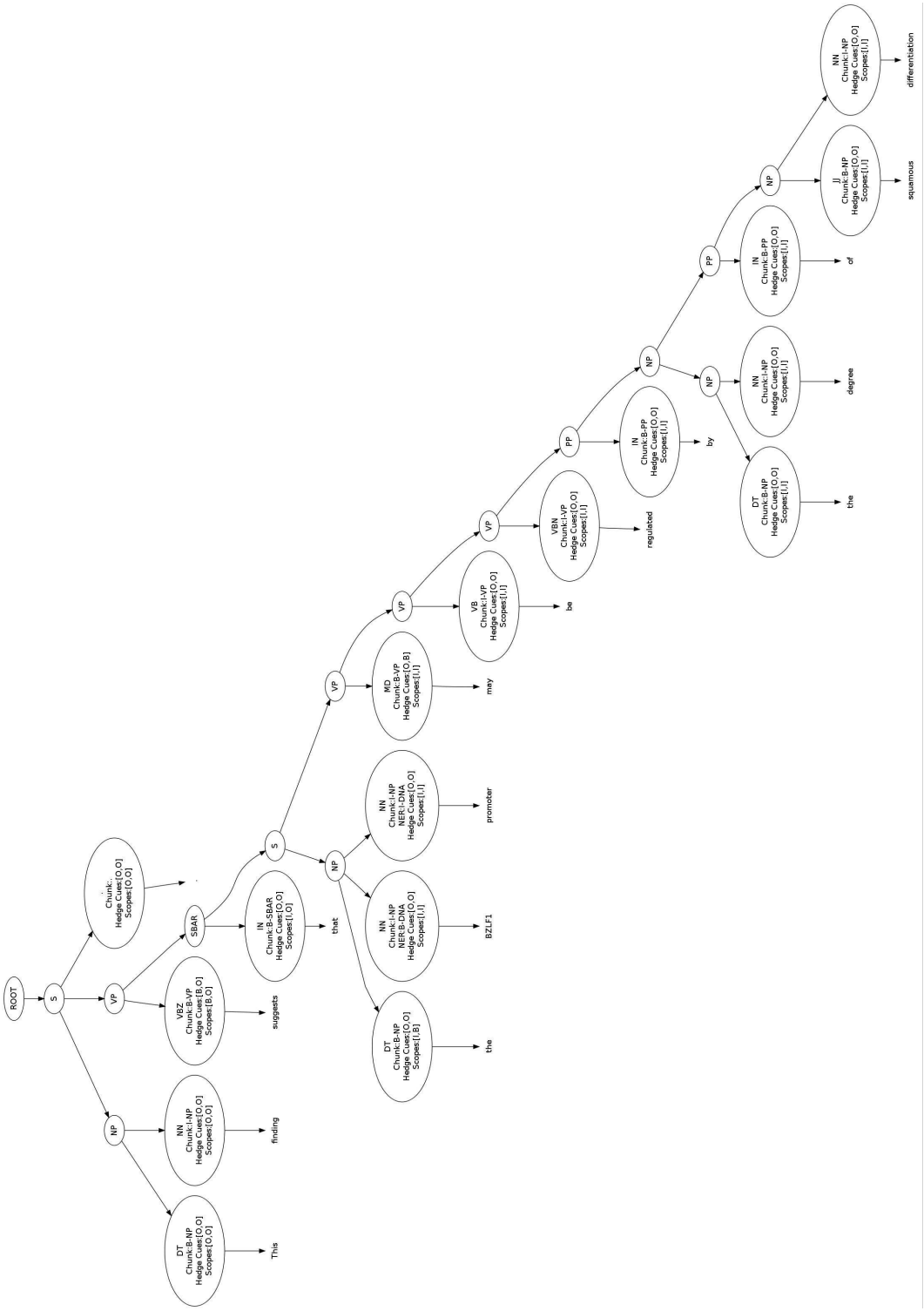

Figure 4.2: Parsing information augmented with lexical, hedging and syntactic features

The addition of new information to the corpus comes at the cost of introducing analysis errors. In this work, errors came from two sources: tagging and syntactic analysis. Studying (and solving) errors that were introduced during the process is a pending and cost-consuming task (which should include the work of linguists and domain specialists). Using a domain-specific tagger and passing this tagging information to the parser, we expected to minimize these errors. The GENIA tagger has reported accuracy of 0.96-0.98 on the domain (Pyysalo et al., 2006; Tsuruoka et al., 2005), and the Stanford parser presents an F-score of 0.86 (using their own tagging method). Based on this information, we think the tagging and parsing errors introduced still allow for the use of the tagged data to improve performance on supervised learning tasks.

Besides adding information to the original corpus, we tried to provide mechanisms for experts to easily browse the corpus content (including sentences and associated features). To achieve this, we added visualization aids to the corpus: based on the original corpus XML file, and using XSLT and CSS templates, the user could browse the corpus, with hedge and negations cues and their scopes highlighted. Figures 4.3 to 4.6 show some examples of these visualization aids. A tree visualization of the final structure was included, as well as a token-per-token visualization of the lexical and hedging features, and the possibility to examine the original sentence and XML structure.

Document:1851858 Bioscope Genia Event

S31.1	Differentiation-associated expression of the Epstein-Barr virus BZLF1 transactivator protein in oral hairy leukoplakia.	Tree Attributes
S31.2	The BZLF1 protein of Epstein-Barr virus (EBV) is a key immediate-early protein which has been shown to disrupt virus latency in EBV-infected B cells.	Tree Attributes
S31.3	We have generated a monoclonal antibody, BZ1, to BZLF1 which reacts in immunohistology, immunoblotting, and immunoprecipitation and which recognizes both the active, dimeric form and the inactive, monomeric form of the protein.	Tree Attributes
S31.4	Biopsies of oral hairy leukoplakia, an AIDS-associated lesion characterized by high-level EBV replication, were examined by immunohistochemistry using the BZ1 monoclonal antibody.	Tree Attributes
S31.5	A differentiation-associated pattern of BZLF1 expression was observed, BZ1 reacting with nuclei of the upper spinous layer of the lesion.	Tree Attributes
S31.6	This finding **suggests** that the BZLF1 promoter **may** be regulated by the degree of squamous differentiation.	Tree Attributes
S31.7	A comparison of in situ hybridization to EBV DNA and viral capsid antigen staining with BZ1 reactivity **suggested** that BZLF1 expression precedes rampant virus replication.	Tree Attributes
S31.8	The inability to detect EBV in the lower epithelial layers of oral hairy leukoplakia **raises questions** concerning the nature of EBV latency and persistence in stratified squamous epithelium.	Tree Attributes

Figure 4.3: Corpus visualization aids. Hedge cues and their scopes are boxed

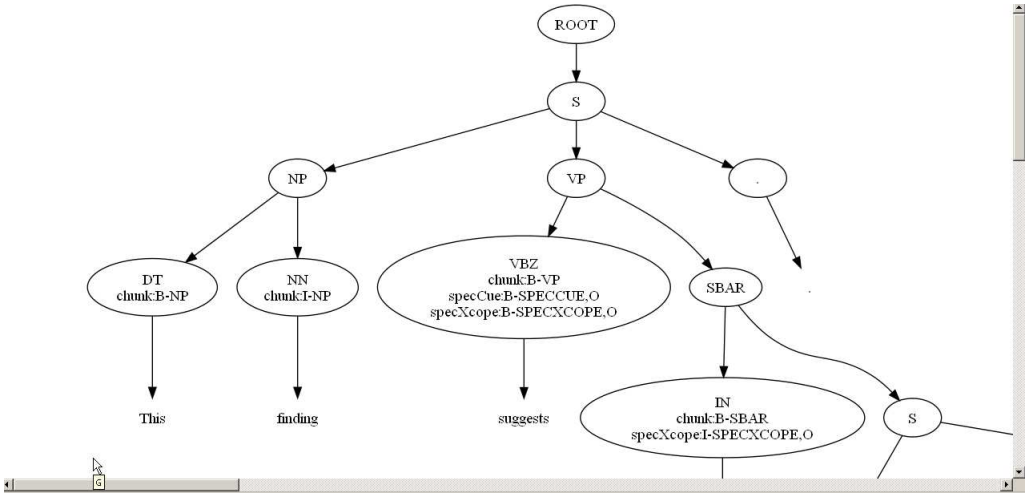

Figure 4.4: Corpus visualization aids. Part of the tree visualization of the sentence

4.3 Training and Evaluation Corpora

Since the methodology proposes the iterative improvement of classifiers based on error analysis, it was particularly important to keep an unseen evaluation corpus until the final classifier was built, allowing us to better evaluate the real performance of the classifier. So we first split the corpus into two sub corpora, one for training and the other one for evaluating performance. To evaluate intermediate classifiers and for parameter tuning, we further divided the training corpus, separating a held-out corpus with about 20% of the sentences of the training corpus, randomly selected. Table 4.4 shows statistics for the three sub corpora.

To evaluate classifier performance, we took a perfect-match approach: we considered an evaluation instance as correctly identified only if every token in the sentence were correctly classified. This means that, for the case of hedge cue identification, we expected every token in the hedge cue to be marked as `B` or `I` while, for the case of scope detection, the scope was considered correctly detected if both the first and last token of the scope were correctly marked.

Classification performance was measured in terms of the traditional figures of precision, recall, and F-score. For the scope detection task, these three numbers coincided, since every False Positive (instances with incorrectly classified scope), implied a False Negative (instances where the correct scope was not identified).

To accurately measure classification performance for the second task, we used the gold standard hedge cues (i.e. those human annotated in the evaluation corpus), instead of the

TOKEN	LEMMA	POS	CHUNK	NE	SPEC-CUE	NEG-CUE	SPEC-XCOPE	NEG-XCOPE
This	This	DT	B-NP	O	[O,O]	[O]	[O,O]	[O]
finding	finding	NN	I-NP	O	[O,O]	[O]	[O,O]	[O]
suggests	suggest	VBZ	B-VP	O	[B-SPECCUE,O]	[O]	[B-SPECXCOPE,O]	[O]
that	that	IN	B-SBAR	O	[O,O]	[O]	[I-SPECXCOPE,O]	[O]
the	the	DT	B-NP	O	[O,O]	[O]	[I-SPECXCOPE,B-SPECXCOPE]	[O]
BZLF1	BZLF1	NN	I-NP	B-DNA	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
promoter	promoter	NN	I-NP	I-DNA	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
may	may	MD	B-VP	O	[O,B-SPECCUE]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
be	be	VB	I-VP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
regulated	regulate	VBN	I-VP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
by	by	IN	B-PP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
the	the	DT	B-NP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
degree	degree	NN	I-NP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
of	of	IN	B-PP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
squamous	squamous	JJ	B-NP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
differentiation	differentiation	NN	I-NP	O	[O,O]	[O]	[I-SPECXCOPE,I-SPECXCOPE]	[O]
.	.	.	O	O	[O,O]	[O]	[O,O]	[O]

Figure 4.5: Corpus visualization aids. Token-per-token visualization

```
− <sentence id="S31.6">
    This finding
  − <xcope id="X31.6.2">
      <cue ref="X31.6.2" type="speculation">suggests</cue>
      that
    − <xcope id="X31.6.1">
        the BZLF1 promoter
        <cue ref="X31.6.1" type="speculation">may</cue>
        be regulated by the degree of squamous differentiation
      </xcope>
    </xcope>
    .
  </sentence>
```

Figure 4.6: Corpus visualization aids. Original XML structure

	Training		Evaluation
	Main	Held out	
#Sentences	7138	1798	2302
Speculative sentences	19%	18%	17%
#Hedge cues	1740	400	524

Table 4.4: Working corpora for training and evaluation

classification results for hedge cue identification.

4.4 Sequential Classification Method

For the two tasks we addressed, we decided to use linear chain Conditional Random Fields (Lafferty et al., 2001; Sha and Pereira, 2003), the state-of-the-art classification method used for sequence supervised learning in many NLP tasks. CRFs are a special case of log-linear models, an extension of logistic regression.

A log-linear model assumes that the probability of any output class y given an example x is

$$p(y|x;w) = \frac{\exp \sum_{j=1}^{J} w_j F_j(x,y)}{Z(x,w)}$$

where each *feature function* $F_j(x,y)$ could be seen as a specific measure of the compatibility between the example x and the output label y, and the corresponding *weight* parameter w_j describes its influence (Elkan, 2013). The function Z in the denominator is a normalizing factor. The learning problem for log-linear models is to calculate the weights w_j associated with each feature (positive weights make y more likely as the true label of x, given everything else fixed).

Conditional Random Fields are a special case of log-linear models used to predict complex labels such as sequences (e.g. multiword hedge cues) from complex input (e.g. sentences seen as a sequence of tokens). Feature functions for linear-chain CRFs are defined in terms of lower level functions, and can depend on the whole input sequence, the current tag y_i and the previous tag y_{i-1}. For example, a CRF feature for hedge cue identification could be: 'the current tag is `B` and the previous tag is `O`, and the current word is 'indicate' and the next word is 'that'. During training, this feature would probably get a high weight since it occurs frequently in the

Token	Word	Lemma	POS	Chunk	NER	Hedge cue
1	This	This	DT	B-NP	O	O
2	finding	finding	NN	I-NP	O	O
3	suggests	suggest	VBZ	B-VP	O	B
4	that	that	IN	B-SBAR	O	O
5	the	the	DT	B-NP	O	O
6	BZLF1	BZLF1	NN	I-NP	B-DNA	O
7	promoter	promoter	NN	I-NP	I-DNA	O
8	may	may	MD	B-VP	O	B
9	be	be	VB	I-VP	O	O
10	regulated	regulate	VBN	I-VP	O	O
11	by	by	IN	B-PP	O	O
12	the	the	DT	B-NP	O	O
13	degree	degree	NN	I-NP	O	O
14	of	of	IN	B-PP	O	O
15	squamous	squamous	JJ	B-NP	O	O
16	differentiation	differentiation	NN	I-NP	O	O
17	.	.	.	O	O	O

Table 4.5: Initial learning attributes for hedge cue identification

training corpus.

For the different scenarios we built we generally used first order Markov CRFs, i.e. let feature functions depend on the sentence input attributes and the current and previous output tags, as the feature tables will show. We used this models because they showed better performance during initial experiments. When we evaluated results we also measured the impact of using plain Markov 0 CRF (i.e. only taking into account the current output tag for feature functions).

4.5 Hedge Cue Identification

To identify the presence of hedge cues in sentences we start with a sequential classifier based on linear-chain CRFs. The input format is the standard learning format used in the CoNLL Shared Task 2006 (Buchholz and Marsi, 2006), where sentences are separated by a blank line and fields are separated by a single tab character. A sentence consists of tokens, each one starting in a new line. Each token field represents a learning attribute. Table 4.5 shows a learning instance obtained from the sentence in example 1.14. Every attribute was already present in the consolidated structure described in the previous section.

The initial attributes we considered for building our classifiers were surface form of each word, lemma, POS tag, chunk tag and the target IOB class marking a hedge cue.

In the previous example, we can see that the words ‘suggest’ and ‘may’ have a `B` value for

Token	Word	Lemma	POS	Chunk	NER	Hedge cue
1	Cotransfection	Cotransfection	NN	B-NP	O	O
2	studies	study	NNS	I-NP	O	O
3	with	with	IN	B-PP	O	O
4	this	this	DT	B-NP	O	O
5	cDNA	cDNA	NN	I-NP	B-DNA	O
6	indicate	indicate	VBP	B-VP	O	B
7	that	that	IN	B-SBAR	O	I
8	it	it	PRP	B-NP	O	O
9	can	can	MD	B-VP	O	B
10	repress	repress	VB	I-VP	O	O
11	basal	basal	JJ	B-NP	O	O
12	promoter	promoter	NN	I-NP	O	O
13	activity	activity	NN	I-NP	O	O
14	.	.	.	O	O	O

Table 4.6: Multiple token cues

the hedge cue attribute, indicating they both start a hedge cue. When hedge cues span multiple tokens, words other than the first token of the cue are marked with `I`: Table 4.6 shows the learning instance for example 4.3: in this case the word 'indicate' has value `B` for the hedge cue while the next token indicate that the cue continues, forming the 'indicate that' hedge cue. For the special case of noncontiguous hedge cues, we used a distinctive `D` tag to indicate that the second token is part of the same hedge cue than the first, distinguishing this case from those where two hedge cues exist in the same sentence (Morante, 2012).

The CRF learning method takes a set of learning instances and (using a statistical approach) trains a classifier to assign to each sentence a sequence of classes that maximize the probability of matching their values in the training corpus, using a maximum likelihood approach. For the hedge cue learning task, the target class was the hedge cue feature, for each sentence token.

Our baseline classifier used as learning attributes just the surface form of each word and the surface form of the two previous and following tokens; the bigrams and trigrams composed of the previous and current word, and of the current and following words. This baseline classifier (evaluated on the held-out corpus) achieved a 0.955 precision and 0.834 F-score.

After a grid search on different configurations of surface forms, lemmas and POS tags, we found (somewhat surprisingly) that the best precision/recall trade off was obtained just using a window of size two of unigrams of surface forms, lemmas and tokens. The slightly worse precision than the baseline classifier was compensated by an improvement of about six points in recall, achieving an F-score of 0.869. Even adding chunk information was unsuccessful: both precision and recall were slightly worse than those of the simpler classifier just mentioned. So,

Tokens	TP	FN	FP	Occurrences in Training Corpus	
				Marked As Hedge Cue	Not Marked
or	3	12	0	92	836
could	2	8	0	39	89
potential	4	3	3	25	89
either	0	5	0	20	123
can	1	5	0	37	319
unknown	0	3	1	11	27
putative	5	2	2	27	21
hypothesis	0	1	2	10	9
indicating	2	1	2	8	55
appeared	2	0	3	18	11
and/or	0	2	0	6	31
not	0	2	0	4	966
not clear	0	2	0	2	2
considered	0	2	0	1	10
if	1	2	0	10	27
potentially	1	2	0	10	7
apparently	1	2	0	2	12
indicate	0	1	1	3	196

Table 4.7: Learning statistics for hedge cues misclassified more than once by the initial classifier. TP stands for True Positives (i.e. correctly identified hedge cues), FN for False Negatives and FP for False Positives. The last two columns show the number of times the tokens appeared in the training corpus as a hedge cue and the number of times they did not.

we decided to keep that classifier as our initial hedge cue identification strategy.

4.5.1 Adding External Knowledge and Co-occurrences

After the initial classifier was built, and following the methodology, we analysed classification errors on the held-out corpus, to see whether we could gather there additional information that could help us to improve classifier performance. We aimed to somehow characterize the most common errors to infer their causes and act on them.

We first counted, for each actual or guessed hedge cue, how many times classifier was unable to predict them (false negatives) and how many times it was wrongly guessed (false positives). In both cases, we also wished to know which of the hedge cue tokens appeared in the training data and how many times they did so acting as a hedge cue. Table 4.7 shows this information for the most frequently misclassified hedge cues in the held-out corpus.

We can see that most errors are false negatives, i.e. the classifier did not mark the occurrence of the hedge cue as such. The reason seems clear when we look at the two last columns of the table: most of these token sequences appeared much more frequently in the corpus not act-

about, almost, apparent, apparently, appear, appeared, appears, approximately, around, assume, assumed, certain amount, certain extent, certain level, claim, claimed, could, doubt, doubtful, essentially, estimate, estimated, feel, felt, frequently, from our perspective, generally, guess, in general, in most cases, in most instances, in our view, indicate, indicated, largely, likely, mainly, may, maybe, might, mostly, often, on the whole, ought, perhaps, plausible, plausibly, possible, possibly, postulate, postulated, presumable, probable, probably, relatively, roughly, seems, should, sometimes, somewhat, suggest, suggested, suppose, suspect, tend to, tends to, typical, typically, uncertain, uncertainly, unclear, unclearly, unlikely, usually, would, broadly, tended to, presumably, suggests, from this perspective, from my perspective, in my view, in this view, in our opinion, in my opinion, to my knowledge, fairly, quite, rather x, argue, argues, argued, claims, feels, indicates, supposed, supposes, suspects, postulates

Table 4.8: Complete list of hedges identified in Hyland (1995)

ing as hedge cues, and that correlation was learned by the statistical classifier. The token 'or', for example, behaved as a hedge cue in only about the 10% of its occurrences in the training corpus and that caused that the classifiers never predicted it as a hedge cue.

We also observed that some misclassified hedge cues (a total of 24 cases) co-occurred with other hedge cues, such as those in example 1.14. This behaviour had previously been noted in the literature (Hyland, 1995) and was present in the training data (for 317 cases, a 18% of the 1740 hedge cues).

To incorporate this information to classification, we decided to include three new attributes for learning. The first one marked the membership of some tokens to a list of hedge cues identified by Hyland (1995), and shown in Table 4.8: this attribute had a value of `Y` if the tokens were part of a hedge cue and belonged to the list, and `N` otherwise. The second and third ones indicated if the token appeared somewhere in the training corpus as a hedge and if it co-occurred with another token in the same situation. Table 4.9 shows the new learning instances.

The information of hedge cue cooccurrences may seem redundant: after all, the classifier can deduce them from training data. But we must take into account that linear-chain CRF (the method we used for building our classification models), for performance reasons, generally consider only a small window of neighbour tokens from the current word, and so can only detect short-length dependencies; the token windows we used as attributes were probably not long enough to include both hedge cues in the sentences. In those cases, their co-occurrence would not be detected by the classifier. When we explicitly marked it, we were adding new information for classification.

After incorporating these attributes, and tuning again for optimizing performance on the held-out corpus, we came with a new classifier that included the attributes listed in Table 4.10[1].

[1]To show classification features for CRF, we follow the notation used in Sha and Pereira (2003), where

Token	Word	Lemma	POS	Hyland Hedge	HC Candidate	Co-occurs with HCC	Hedge cue
1	This	This	DT	N	N	N	O
2	finding	finding	NN	N	N	N	O
3	suggests	suggest	VBZ	Y	Y	Y	B
4	that	that	IN	N	N	N	O
5	the	the	DT	N	N	N	O
6	BZLF1	BZLF1	NN	N	N	N	O
7	promoter	promoter	NN	N	N	N	O
8	may	may	MD	Y	Y	Y	B
9	be	be	VB	N	N	N	O
10	regulated	regulate	VBN	N	N	N	O
11	by	by	IN	N	N	N	O
12	the	the	DT	N	N	N	O
13	degree	degree	NN	N	N	N	O
14	of	of	IN	N	N	N	O
15	squamous	squamous	JJ	N	N	N	O
16	differentiation	differentiation	NN	N	N	N	O
17	.	.	.	N	N	N	O

Table 4.9: New learning attributes for hedge cue identification

We can see that Hyland hedges were not included in the final classifier because they did not improve performance. This has probably happened because the classifier already learned those cues from training data.

This classifier achieved an F-measure of 0.875, an improvement of 0.6 percentage points with respect to the initial classifier performance. A fall of 3 points in precision was compensated with an improvement 3.5 points of recall in the held-out corpus as it was expected. In the following chapter, we show that this improvement also hold on the evaluation corpus. Table 4.11 summarizes the results presented in this section.

4.6 Scope Detection

The second task we addressed was to know, given a sentence where one or more hedge cues were identified, which sentence span they affected. Recalling example 1.14, once we had identified the 'suggest' and 'may' hedge cues, we expected the classifier to learn that, unless in the Bioscope corpus, the scope of 'suggest' was the part of the sentence that started with the word itself, and ended at the end of the sentence (modulo the final period), and that the scope of 'may' was included within it, matching the passive voice clause headed by the hedge cue.

$q(y_{i-1}, y_i)$ is a predicate on output labels and current position i, and $p(\mathbf{x}, i)$ is a predicate on the input sequence

$q(y_{i-1}, y_i)$	$p(x, i)$
$y_i = y$ $y_i = y, y_{i-1} = y'$	true
$y_i = y$	$w_i = w$
	$w_{i-1} = w$
	$w_{i+1} = w$
	$w_{i-2} = w$
	$w_{i+2} = w$
	$w_{i-3} = w$
	$w_{i+3} = w$
	$w_{i-4} = w$
	$w_{i+4} = w$
	$l_i = l$
	$l_{i-1} = l$
	$l_{i+1} = l$
	$l_{i-2} = l$
	$l_{i+2} = l$
	$t_i = t$
	$t_{i-1} = t$
	$t_{i+1} = t$
	$t_{i-2} = t$
	$t_{i+2} = t$
	$h_i = h$
	$h_{i-1} = h$
	$h_{i+1} = h$
	$h_{i-2} = h$
	$h_{i+2} = h$
	$c_i = c$
	$c_{i-1} = c$
	$c_{i+1} = c$
	$c_{i-2} = c$
	$c_{i+2} = c$

Table 4.10: Attributes for the improved classifier for hedge cue identification. w_i represents the current word, l_i its lemma, t_i its POS tag, h_i is a y/n attribute that indicates if it is a hedge cue candidate, while c_i similarly indicates if it cooccurs with a hedge cue candidate. y_i is the current token class

Configuration	P	R	F1
Baseline	**0.955**	0.74	0.834
Initial Classifier	0.944	0.805	0.869
Improved Classifier	0.913	**0.84**	**0.875**

Table 4.11: Classification performance on the held out corpus for hedge cue detection.

Token	Word	Lemma	POS	Hedge cue	Scope
1	This	This	DT	O	O
2	finding	finding	NN	O	O
3	suggests	suggest	VBZ	B	F
4	that	that	IN	O	O
5	the	the	DT	O	O
6	BZLF1	BZLF1	NN	O	O
7	promoter	promoter	NN	O	O
8	may	may	MD	O	O
9	be	be	VB	O	O
10	regulated	regulate	VBN	O	O
11	by	by	IN	O	O
12	the	the	DT	O	O
13	degree	degree	NN	O	O
14	of	of	IN	O	O
15	squamous	squamous	JJ	O	O
16	differentiation	differentiation	NN	O	L
17	.	.	.	O	O

Table 4.12: Learning Instance #1 for sentence 1.14

As the previous example showed, there could be more than one hedge cue in the same sentence, each of them inducing a different scope. In this case, we considered as learning instances the pairs ⟨sentence, hedge cue start position⟩, for each hedge cue in the sentence (Morante and Daelemans, 2009). No training instances were generated for corpus sentences where no hedge cue was present. For example, for sentence 1.14 we generated two learning instances:

- ⟨This finding {suggests that the BZLF1 promoter may be regulated by the degree of squamous differentiation}., 3⟩
- ⟨This finding suggests that {the BZLF1 promoter may be regulated by the degree of squamous differentiation}., 8⟩

Similar to the case of hedge cue recognition, the usual approach for this task is to consider it as a sequential labelling task: each instance is converted to a sequence of tokens with their attributes, and a class is assigned to each token in a `FOL` format (tagging as `F` the first token of the scope and as `L` the last one). Tables 4.12 and 4.13 show the learning instances for the aforementioned sentence, using the set of attributes of the baseline classifier, explained in the following section.

While generating the learning instances was quite simple (we identified in the learning corpus a hedge cue and generated an instance with the sentence where the scope was marked),

Token	Word	Lemma	POS	Hedge cue	Scope
1	This	This	DT	O	O
2	finding	finding	NN	O	O
3	suggests	suggest	VBZ	O	O
4	that	that	IN	O	O
5	the	the	DT	O	F
6	BZLF1	BZLF1	NN	O	O
7	promoter	promoter	NN	O	O
8	may	may	MD	B	O
9	be	be	VB	O	O
10	regulated	regulate	VBN	O	O
11	by	by	IN	O	O
12	the	the	DT	O	O
13	degree	degree	NN	O	O
14	of	of	IN	O	O
15	squamous	squamous	JJ	O	O
16	differentiation	differentiation	NN	O	L
17	.	.	.	O	O

Table 4.13: Learning Instance #2 for sentence 1.14

we had also to take into account the special cases of discontinuous hedge cues. In these cases, just one learning instance was generated, as Table 4.14 shows.

These learning instances were generated from the same training, held-out and evaluation corpora described in the previous section. The number of instances for each corpus is shown in Table 4.15. We can see that there are 40 learning instances less in the corpus than the number of hedge cues, corresponding to non contiguous hedge cues that induced just one scope as we explained in the previous paragraph.

4.6.1 Baseline Classifier

What we considered as learning attributes for our baseline classifier were the same lexical attributes (word, lemma, and part-of-speech class) we used for hedge cue identification, adding the hedge cue attribute learned in the previous step; all four attributes were considered in a window of size 5, centered in the current token. As we have previously mentioned, every attribute considered for learning was already present in the same consolidated structure we used for hedge cue identification. The learning method used was, again, Conditional Random Fields. The previously shown example instance for example 1.14 corresponded to this classifier configuration.

We also included (as part of the baseline classifier) a set of postprocessing rules that were

Token	Word	Lemma	POS	Hedge cue	Scope
...					
18	PMA	PMA	NN	O	O
19	plus	plus	CC	O	O
20	either	either	CC	B	F
21	lectin	lectin	NN	O	O
22	or	or	CC	D	O
23	antibody	antibody	NN	O	O
24	to	to	TO	O	O
25	the	the	DT	O	O
26	CD3	CD3	NN	O	O
27	or	or	CC	O	O
28	CD28	CD28	NN	O	O
29	surface	surface	NN	O	O
30	molecules	molecule	NNS	O	L
31	.	.	.	O	O

Table 4.14: Learning Instance including discontinuous hedge cues

	Training		Evaluation
	Main	Held out	
#Scopes	1710	393	521

Table 4.15: Training and evaluation instances for scope detection

fired when one (or both) scope limits could not be identified: when we transformed the task of scope detection into a sequential classification one and used a `FOL` format, it was possible that not exactly one `F` and one `L` class were predicted for an instance sentence. Table 4.16 shows the correct and predicted scope tags for a sentence in the corpus: the classifier predicted two tokens as `F`. In this case, we knew for sure that the correct scope could not be identified, and should try to backoff to a simpler guess.

The set of rules we used to modify the classifier results on evaluation data to correct those cases were a slightly modified form of the ones presented in (Morante and Daelemans, 2009), and are enumerated in Table 4.17.

This baseline classifier, applied to our held-out corpus, obtained a F-measure value of 0.664.

Token	Word	Lemma	POS	Hedge cue	Scope	Predicted	Corrected
1	Deletion	Deletion	NN	O	O	F	O
2	or	or	CC	O	O	O	O
3	substitution	substitution	NN	O	O	O	O
4	of	of	IN	O	O	O	O
5	a	a	DT	O	O	O	O
6	putative	putative	JJ	B	F	F	F
7	NF-kappaB	NF-kappaB	NN	O	O	O	O
8	binding	binding	NN	O	O	O	O
9	site	site	NN	O	O	O	O
10	identified	identify	VBN	O	O	O	O
11	in	in	IN	O	O	O	O
12	the	the	DT	O	O	O	O
13	bcl-x	bcl-x	NN	O	O	O	O
14	promoter	promoter	NN	O	L	O	O
15	significantly	significantly	RB	O	O	O	O
16	decreased	decrease	VBD	O	O	O	O
17	Tax-induced	Tax-induced	JJ	O	O	O	O
18	transactivation	transactivation	NN	O	O	L	L
19	.	.	.	O	O	O	O

Table 4.16: Learning Instance, including postprocessing to correct badly formed scopes.

- If exactly one `F` was predicted, but no token was classified as `L`, predict as both scope start and end the token predicted as `F` (producing a one-word scope).
- If exactly one `L` was predicted, but no token was classified as `F`, predict the first token of the hedge cue as scope start and let the scope end match the predicted `L`.
- If exactly one `F` but more than one `L` were predicted, use the first `L` after the `F` as scope end (if that is not possible, build a one-word scope using the token for `L`).
- If exactly one `L` but more than one `F` were predicted, use the first token of the hedge cue as scope start.
- If no `F` and more than one `L` were predicted, predict the first token of the hedge cue as the scope start, and predict as scope end the first `L` after the hedge cue (if that is not possible, build a one-word scope using the token for the first `L`).
- If more than one `F` and more than one `L` were predicted, use the first `F` as scope start, and the first `L` as scope end.
- If neither `F` nor `L` were predicted, let the scope include only the words in the hedge cue.
- Finally, if for the same sentence there exist two or more overlapping scopes, shrink them to include only the words in their respective hedge cues.

Table 4.17: Postprocessing rules for the initial classifier

```
(ROOT
  (S
    (NP (DT This) (NN finding))
    (VP (VBZ suggests)
      (SBAR (IN that)
        (S
          (NP (DT the) (NN BZLF1) (NN promoter))
          (VP (MD may)
            (VP (VB be)
              (VP (VBN regulated)
                (PP (IN by)
                  (NP
                    (NP (DT the) (NN degree))
                    (PP (IN of)
                      (NP (JJ squamous) (NN differentiation))))))))))
    (. .)))
```

Figure 4.7: Parse for sentence 1.14. The scope of the hedge cue is shown in bold

4.6.2 Iteration 1: Adding Syntax Information

We observed that most scopes were associated with the sentence syntactic constituents, particularly those that included the hedge cue. For example, looking at the Bioscope corpus annotation guidelines we recalled that the scope of a verb such as 'suggest' corresponded to the parent component of the hedge cue in the parse tree, as Figure 4.7 shows for example 1.14.

To improve classification, we included as a learning attribute a knowledge rule stating that the scope of the hedge cue was the syntactic scope of the parent of the hedge cue (i.e. the parent in the parse tree of its first word), modulo final periods. Or, in terms of attribute values:

`in_hc_parent_scope`	=	`F` when the token is the first word of the parent of the hedge cue
	=	`L` when the token is the last word of the parent of the hedge cue
	=	`O` otherwise

Since this criterion did not hold for every part-of-speech (and not even of every use of the verb, as, for example, passive voice construction or raising verbs cases show), we also included as a learning attribute the part-of-speech of the hedge cue parent. Tables 4.18 and 4.19 show the classifier attributes for both hedge cues in sentence 1.14. We can see that in the case of 'suggest', the `in_hc_parent_scope` rule matched the cue scope, while that did not happen

Token	Word	Lemma	POS	HC	PPOS	in-PScope	Scope
1	This	This	DT	O	VP	O	O
2	finding	finding	NN	O	VP	O	O
3	suggests	suggest	VBZ	B	VP	F	F
4	that	that	IN	O	VP	O	O
5	the	the	DT	O	VP	O	O
6	BZLF1	BZLF1	NN	O	VP	O	O
7	promoter	promoter	NN	O	VP	O	O
8	may	may	MD	O	VP	O	O
9	be	be	VB	O	VP	O	O
10	regulated	regulate	VBN	O	VP	O	O
11	by	by	IN	O	VP	O	O
12	the	the	DT	O	VP	O	O
13	degree	degree	NN	O	VP	O	O
14	of	of	IN	O	VP	O	O
15	squamous	squamous	JJ	O	VP	O	O
16	differentiation	differentiation	NN	O	VP	L	L
17	.	.	.	O	VP	O	O

Table 4.18: Learning Instance after Iteration 1 for sentence 1.14, hedge cue 'suggest'

in the case of the hedge cue 'may'. We expected that the classifier could confirm or discard a correlation between the knowledge rule and the scope, based on training data.

The use of `FOL` classes to identify the first and last token of the hedge scope introduced a possible error source into the classification: the classifier could correctly learn the first but not the last token of the sequence. We would like to identify the event 'the hedge scope coincides with the parent constituent scope' rather than just rely on the sequential learning method to detect scope borders. To achieve this without changing the learning paradigm, we introduced what we called an X-rule in the Methodology section: every time a hedge scope coincided with the constituent scope in the learning corpus, we classified the first token of the scope with a special `X` class. After this modification, the classifier could learn the particular class when there existed strong statistical evidence not only that certain token was the first in the hedge scope, but also that this was due to its position in the syntax tree. Table 4.20 shows the modified version of the learning instance for example 1.14, hedge cue 'suggest'.

During the evaluation phase, every evaluation instance was tagged in a new `{F/X}OL` format (identical to that used for training); since our final problem involved `FOL` marking we postprocessed classification results, changing every `X` with `F`, and also forcing the `L` class to coincide with the token whose `in_hc_parent_scope` attribute was `L`. Table 4.21 shows an hypothetical evaluation instance, with the original, guessed and postprocessed scope.

Token	Word	Lemma	POS	HC	PPOS	in-PScope	Scope
1	This	This	DT	O	VP	O	O
2	finding	finding	NN	O	VP	O	O
3	suggests	suggest	VBZ	O	VP	O	O
4	that	that	IN	O	VP	O	O
5	the	the	DT	O	VP	O	F
6	BZLF1	BZLF1	NN	O	VP	O	O
7	promoter	promoter	NN	O	VP	O	O
8	may	may	MD	B	VP	F	O
9	be	be	VB	O	VP	O	O
10	regulated	regulate	VBN	O	VP	O	O
11	by	by	IN	O	VP	O	O
12	the	the	DT	O	VP	O	O
13	degree	degree	NN	O	VP	O	O
14	of	of	IN	O	VP	O	O
15	squamous	squamous	JJ	O	VP	O	O
16	differentiation	differentiation	NN	O	VP	L	L
17	.	.	.	O	VP	O	O

Table 4.19: Learning Instance after Iteration 1 for sentence 1.14, hedge cue 'may'

Token	Word	Lemma	POS	HC	PPOS	in-PScope	Scope
1	This	This	DT	O	VP	O	O
2	finding	finding	NN	O	VP	O	O
3	suggests	suggest	VBZ	B	VP	F	X
4	that	that	IN	O	VP	O	O
5	the	the	DT	O	VP	O	O
6	BZLF1	BZLF1	NN	O	VP	O	O
7	promoter	promoter	NN	O	VP	O	O
8	may	may	MD	O	VP	O	O
9	be	be	VB	O	VP	O	O
10	regulated	regulate	VBN	O	VP	O	O
11	by	by	IN	O	VP	O	O
12	the	the	DT	O	VP	O	O
13	degree	degree	NN	O	VP	O	O
14	of	of	IN	O	VP	O	O
15	squamous	squamous	JJ	O	VP	O	O
16	differentiation	differentiation	NN	O	VP	L	L
17	.	.	.	O	VP	O	O

Table 4.20: Learning Instance after Iteration 1 for sentence 1.14, hedge cue 'suggest', using X-rule attributes for parent scope

#	Lemma	POS	HC	PPOS	in-PScope	Scope	Guessed	Predicted Scope
1	Take	VBN	O	VP	O	O	O	O
2	together	RB	O	VP	O	O	O	O
3	,	,	O	VP	O	O	O	O
4	these	DT	O	VP	O	O	O	O
5	result	NNS	O	VP	O	O	O	O
6	suggest	VBP	B	VP	F	F	X	F
7	that	IN	O	VP	O	O	O	O
8	NF-kappa	NN	O	VP	O	O	O	O
9	B	NN	O	VP	O	O	O	O
10	play	VBZ	O	VP	O	O	O	O
11	a	DT	O	VP	O	O	O	O
12	crucial	JJ	O	VP	O	O	O	O
13	role	NN	O	VP	O	O	O	O
14	in	IN	O	VP	O	O	O	O
15	ensure	VBG	O	VP	O	O	O	O
16	the	DT	O	VP	O	O	O	O
17	differentiation	NN	O	VP	O	O	O	O
18	and	CC	O	VP	O	O	O	O
19	survival	NN	O	VP	O	O	O	O
20	of	IN	O	VP	O	O	O	O
21	thymocyte	NNS	O	VP	O	O	L	O
22	in	IN	O	VP	O	O	O	O
23	the	DT	O	VP	O	O	O	O
24	early	JJ	O	VP	O	O	O	O
25	stage	NNS	O	VP	O	O	O	O
26	of	IN	O	VP	O	O	O	O
27	their	PRP$	O	VP	O	O	O	O
28	development	NN	O	VP	L	L	O	L
29	.	.	O	VP	O	O	O	O

Table 4.21: Evaluation using X-rule

$q(y_{i-1}, y_i)$	$p(x, i)$
$y_i = y$ $y_i = y, y_{i-1} = y'$	true
	$hc_i = hc$
	$hc_{i-1} = hc$
	$hc_{i+1} = hc$
	$hc_{i-2} = hc$
	$hc_{i+2} = hc$
	$tp_i = tp$
	$ps_i = ps$
	$ps_{i-1} = ps$
	$ps_{i+1} = ps$
	$ps_{i-2} = ps$
$y_i = y$	$ps_{i+2} = ps$
	$t_i = t$
	$t_{i-1} = t$
	$t_{i+1} = t$
	$t_{i-2} = t$
	$t_{i+2} = t$
	$l_i = l$
	$l_{i-1} = l$
	$l_{i+1} = l$
	$l_{i-2} = l$
	$l_{i+2} = l$

Table 4.22: Attributes for the classifier after Iteration 1 for scope detection. hc_i represents the BIO tag of the current word, l_i its lemma, t_i its POS tag, tp_i is the POS-tag of the Hedge Cue parent in the syntax tree. ps_i is the attribute that marks the yield of the parent of the hedge cue in the syntax tree (valued `F` if the token is the first word of the scope, `L` if it is the last one, and `O` otherwise).

After a grid search for the best attributes and fine-tuning learning parameters (w.r.t performance on the held-out corpus), we built a the classifier using the configuration shown in table 4.22. Using this configuration (keeping the same postprocessing rules and adding the X-rule postprocessing), performance improved by more than 4 points (for a F-score of 0.705 in the held-out corpus). Table 4.23 compares these results with the baseline classifier.

4.6.3 Iteration 2: Adding Ancestors in the Syntax Tree

After the previous iteration, we elaborated a list of the 116 errors the classifier had committed in the held-out corpus and tried to guess why it had been wrong. As we have already said, this method could be applied because attributes for learning corresponded quite directly with a few observable properties related to the linguistic phenomenon we aimed to study. The different possible causes for misclassification we could identify were the following:

Configuration	F	New errors	Solved errors
Baseline	0.664	-	-
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	**0.705**	5	11

Table 4.23: Classification performance on the held-out corpus for scope detection, after iteration #1. The number of new and solved error are relative to the previous classifier in the table

1. Problems with scope selection: the guessed scope coincided with the scope of one of the ancestors of the hedge cue in the parsing tree, other than the parent. For example, in sentence 4.5 the hedge scope matched the grandparent constituent (the clause that starts with 'this expectation...'), but the classifier selected the verb phrase that starts with 'may', which was the parent constituent (until now, the only syntax constituent it was aware of)

   (4.5) While {this expectation [may be realized in some cases]}, we have not found evidence for it.[1]

2. Differences between hedge scope and constituent scope: this error was produced when the hedge scope did not include certain constituents present in the syntactic scope, or vice versa. In sentence 4.6, for example, the hedge scope does not include either the conjunction phrase 'as well as' for the noun phrase 'their fate on cell activation', but the clause containing the hedge cue in the generated parse tree does (probably due to parsing errors).

   (4.6) We therefore investigated [{whether NF-kappaB/Rel proteins are expressed in human neutrophils}, as well as their fate on cell activation].

3. Problems identifying `F`, `L` tokens, or both. In these cases, the classifier failed to identify exactly one `F` and one `L` token, and the postprocessing rules did not correctly predict the correct scopes.

4. Finally, there were cases that looked like annotation problems (i.e. they did not obey the annotation guidelines) or errors induced by tagging or parsing errors. These errors were rare, so we decided not to study them.

[1] We mark the predicted scope with square brackets when it does not coincide with the golden scope

#	Lemma	POS	HC	PPOS	GPPOS	GGPPOS	in-PS	in-GPS	in-GPPS	Scope
1	This	DT	O	VP	S	SBAR	O	O	O	O
2	finding	NN	O	VP	S	SBAR	O	O	O	O
3	suggest	VBZ	O	VP	S	SBAR	O	O	O	O
4	that	IN	O	VP	S	SBAR	O	O	F	O
5	the	DT	O	VP	S	SBAR	O	F	O	Y
6	BZLF1	NN	O	VP	S	SBAR	O	O	O	O
7	promoter	NN	O	VP	S	SBAR	O	O	O	O
8	may	MD	B	VP	S	SBAR	F	O	O	O
9	be	VB	O	VP	S	SBAR	O	O	O	O
10	regulate	VBN	O	VP	S	SBAR	O	O	O	O
11	by	IN	O	VP	S	SBAR	O	O	O	O
12	the	DT	O	VP	S	SBAR	O	O	O	O
13	degree	NN	O	VP	S	SBAR	O	O	O	O
14	of	IN	O	VP	S	SBAR	O	O	O	O
15	squamous	JJ	O	VP	S	SBAR	O	O	O	O
16	differentiation	NN	O	VP	S	SBAR	L	L	L	L
17	.	.	O	VP	S	SBAR	O	O	O	O

Table 4.24: Learning Attributes for Iteration 2 for sentence 1.14, hedge cue 'may'. PPOS, GPPOS and GGPOS are the part-of-speech tags of the parent, grandparent and great-grandparent of the hedge cue, and the three following columns show their scopes. The last column includes the scope of the hedge cue

In this iteration, we decided to address the first type of errors, and we added the same syntactic information for hedge cue grandparents and great-grandparents in the syntax tree, and their part-of-speech tags, exactly the same way we did in the previous section. We fed the classifier with all of them, and let it select which scope to use, depending on the remaining attributes. Table 4.24 shows the attributes for example 1.14 and hedge cue 'may', where the hedge scope coincides with the grandparent constituent. We also included the X-rules for both new knowledge rules (with `Y` and `Z` tags for grandparent and great-grandparent, respectively).

The best classifier configuration for these attributes is shown in table 4.25. We only included X-rules for parent and grandparent (adding great-grandparent actually *reduced* performance on the held-out corpus, probably due to the small number of cases in the learning corpus where the scope matched this constituent).

Table 4.26 updates the evaluation results on the training corpus, comparing them with previous iterations: we can see that performance again improved, and that new rules where very precise, adding only 12 new errors, only one third of the 36 classification errors they solved.

$q(y_{i-1}, y_i)$	$p(x, i)$
$y_i = y$ $y_i = y, y_{i-1} = y'$	true
$y_i = y$	$hc_i = hc$
	$hc_{i-1} = hc$
	$hc_{i+1} = hc$
	$hc_{i-2} = hc$
	$hc_{i+2} = hc$
	$tp_i = tp$
	$tgp_i = tgp$
	$tggp_i = tggp$
	$ps_i = ps$
	$ps_{i-1} = ps$
	$ps_{i+1} = ps$
	$ps_{i-2} = ps$
	$ps_{i+2} = ps$
	$gps_i = gps$
	$gps_{i-1} = gps$
	$gps_{i+1} = gps$
	$gps_{i-2} = gps$
	$gps_{i+2} = gps$
	$ggps_i = gps$
	$ggps_{i-1} = ggps$
	$ggps_{i+1} = ggps$
	$ggps_{i-2} = ggps$
	$ggps_{i+2} = ggps$
	$t_i = t$
	$t_{i-1} = t$
	$t_{i+1} = t$
	$t_{i-2} = t$
	$t_{i+2} = t$
	$l_i = l$
	$l_{i-1} = l$
	$l_{i+1} = l$
	$l_{i-2} = l$
	$l_{i+2} = l$

Table 4.25: Attributes for the classifier after Iteration 2 for scope detection. hc_i represents the BIO tag of the current word, l_i its lemma, t_i its POS tag, tp_i, tgp_i and $tggp_i$ are the POS tags of the ancestors of the hedge cue in the syntax tree, while ps_i, gps_i and $ggps_i$ are the attributes that mark their yields in the syntax tree

Configuration	F	New errors	Solved errors
Baseline	0.664	-	-
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	0.705	5	11
Iteration #2	**0.740**	12	36

Table 4.26: Classification performance on the held-out corpus for scope detection, after iteration #2.

4.6.4 Iteration 3: Adjusting Ancestor Scopes

Until this iteration we had assumed that syntactic scopes matched hedge scopes, i.e. that we could find an ancestor of the hedge cue in the syntax tree whose scope matched the scope of the hedge cue. In the previous section, by studying classification errors, we found that this assumption was not always true. In this iteration we aimed to see whether this was a general problem, and if we could adjust constituent scopes to achieve concordance.

We first studied, for every hedge cue in the training corpus, if its hedge scope coincided with the syntactic scope of one of its ancestors, resulting from the sentence parsing. We found that for about 80% of the hedge cues this was actually true. We can see that, for example, the scopes of the modal verb 'may' (the most common hedge in the corpus) did matched the parent scope for 83% of the learning instances, while in 8% of them coincided with the grandparent scope and there were 27 instances (8% of the total number of examples) where the hedge scope differed with every ancestor's syntax scope. Table 4.27 shows these statistics for hedge cues appearing more than ten times in the training corpus.

To improve this matching, we studied the cases of misalignment where they accounted for a greater proportion and greater number of instances. Table 4.28 shows the cases where more than a half of the hedge scopes were not aligned with ancestor scopes, and the number of total appearances of the hedge cue were greater than three.

analysing these misalignment, we found two main causes:

- The scope of the hedge cue included several non-nested scopes, not allowing a characterization of the hedge scope in terms of syntactic scopes. For example, for sentence 4.7, its syntactic analysis (partially depicted in Figure 4.8), shows that the hedge cue scope (shown in red) includes the parent noun phrase of the hedge cue (excluding the initial determiner) and the clause to the right of the grandparent constituent.

  (4.7) Our results lend further support to the {hypothesis that inflammatory and immune responses of monocytes/macrophages may be modulated at the molecular level by signals originating from tissue structural cells such as fibroblasts}.

- The second type of mismatch included the cases where the hedge scope coincided with just a portion of an ancestor scope, excluding some subconstituents. For example 4.8, the scope of the hedge cue 'suggested' was the grandparent clause of the hedge cue (refer to Figure 4.9), but excluding the final clause introduced by the expression 'as seen in'.

  (4.8) It is {suggested that danazol has an anti-estrogenic action to the monocytes through the competition and suppression of estrogen binding sites} as seen in the

Hedge cue	POS	P	GP	GGP	GGGP	No	Total
may	MD	279	28	1	0	27	335
suggest	VBP	190	4	0	0	17	211
indicate_that	VBP	85	1	0	0	12	98
suggesting	VBG	86	0	0	0	2	88
or	CC	34	3	0	1	30	68
appears	VBZ	9	22	4	0	25	60
whether	IN	54	0	0	0	2	56
could	MD	28	6	1	0	6	41
might	MD	35	3	1	0	2	41
suggests	VBZ	40	0	0	0	0	40
can	MD	29	4	0	0	4	37
indicating_that	VBG	33	0	0	0	2	35
indicated_that	VBD	25	0	0	0	4	29
possible	JJ	5	1	1	0	22	29
putative	JJ	4	0	0	0	23	27
likely	JJ	1	4	5	3	11	24
potential	JJ	4	6	0	0	14	24
propose	VBP	23	0	0	0	1	24
suggested	VBD	24	0	0	0	0	24
thought	VBN	2	7	6	2	5	22
suggested	VBN	7	0	5	6	3	21
appear	VBP	3	8	3	0	5	19
either_or	CC	15	0	0	0	4	19
appeared	VBD	2	6	1	0	9	18
seems	VBZ	4	8	1	0	4	17
possibly	RB	3	2	1	1	9	16
indicates_that	VBZ	13	0	0	0	1	14
likely	RB	1	1	0	0	11	13
probably	RB	0	2	0	1	9	12
should	MD	10	1	0	0	1	12
unknown	JJ	0	0	7	0	4	11

Table 4.27: Alignment of hedge scopes with syntactic constituent scopes, for the most common hedge cues. The P, GP, GGP, GGGP columns contain the number of instances where hedge scope coincides with parent, grandparent, great-grandparent, and great-great-grandparent respectively. The No column counts the cases where no match occurred.

Hedge cue	POS	P	GP	GGP	GGGP	No	Total	%
possible	JJ	5	1	1	0	22	29	76
putative	JJ	4	0	0	0	23	27	85
potential	JJ	4	6	0	0	14	24	58
appeared	VBD	2	6	1	0	9	18	50
possibly	RB	3	2	1	1	9	16	56
likely	RB	1	1	0	0	11	13	85
probably	RB	0	2	0	1	9	12	75
appear	VB	1	0	3	1	5	10	50
hypothesis	NN	0	0	0	0	10	10	100
presumably	RB	1	2	0	0	4	7	57
and/or	CC	1	0	0	0	4	5	80
apparent	JJ	0	0	0	0	5	5	100
possibility	NN	0	0	0	0	5	5	100
not_clear	RB	0	2	0	0	2	4	50
not_known	RB	0	2	0	0	2	4	50
seem	VB	0	0	0	0	3	3	100

Table 4.28: Alignment of hedge scopes with syntactic constituent scopes: most common sources of misalignment.

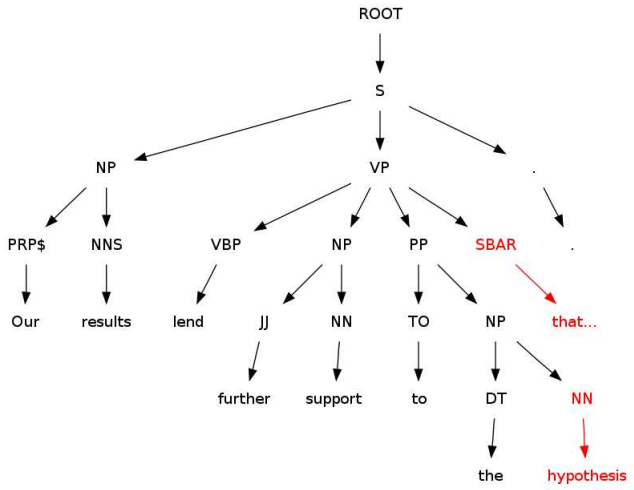

Figure 4.8: Hedge scope for sentence 4.7

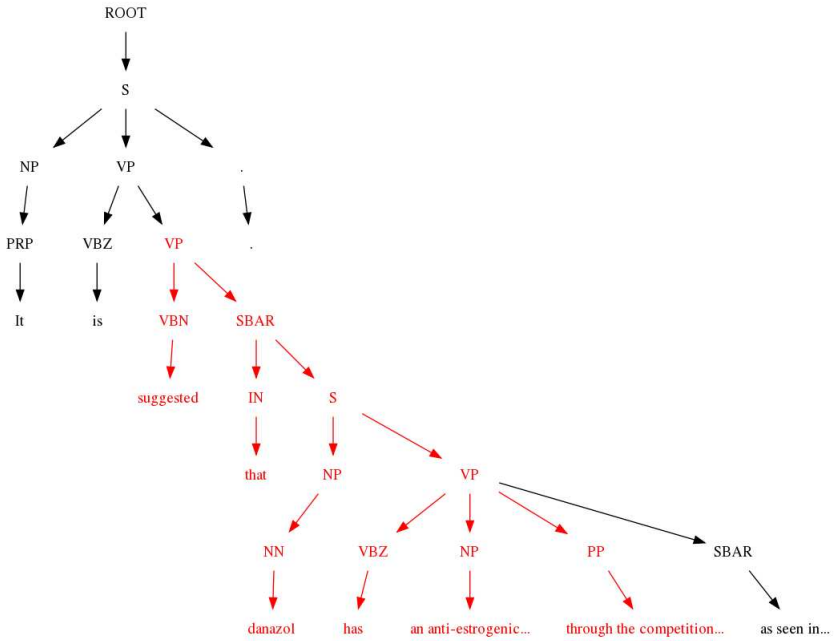

Figure 4.9: Hedge scope for sentence 4.8

estrogen target organ.

We can observe that both examples were probably originated by parsing problems. In the first case, the clause starting with 'that inflammatory...' could be considered within the syntactic scope of the NP clause headed by the noun hedge cue 'hypothesis', while, in the second one, the clause introduced by the preposition 'as' should be a syntactic child of the main sentence clause, and, therefore, not included in the hedge cue scope. Recall from section 4.2.3 that to obtain sentence constituents we used an external parser: therefore, it was not possible (in our working scenario) to correct parsing errors such as those shown in the last examples. However, analysing the relation between hedge scopes and syntactic constituents in the training corpus it was possible for us to derive a series of rules to correct the extracted features and adjust the syntactic scopes based mainly on lexical information. This procedure did not aim to derive a correct parse, but only to produce better features for the task, in certain clearly identified cases. Similar comments could be made to those cases where the syntax constituent was correctly derived, but, due to annotation idiosyncrasies, certain parts should not be included in the hedge scope.

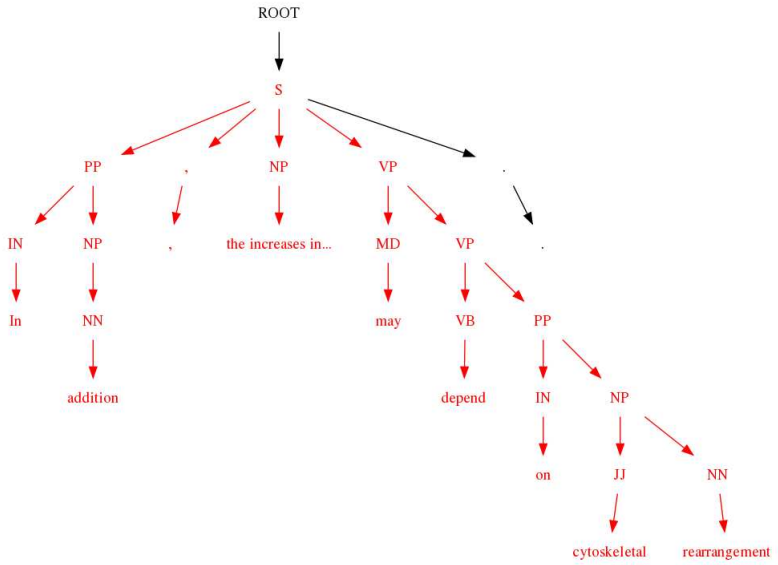

Figure 4.10: Syntax tree for sentence 4.9. The grandparent clause of the hedge cue is shown in red

After looking at several of these mismatches, we modified our definition of 'constituent scope', incorporating the following rules:

- If the constituent is a clause or a verb phrase, exclude from the scope every adverbial or prepositional phrase, as well as every subordinate clause, at the beginning of the constituent. For example 4.9, the scope of the grandparent clause of the hedge cue (which, by the way, does not coincide with the hedge scope), excludes the prepositional phrase 'In addition' (refer to Figure 4.10 for the sentence syntax tree).

  (4.9) In addition, the increases in c-jun, EGR2, and PDGF(B) {may depend on cytoskeletal rearrangement}.

- If the constituent is a clause or a verb phrase and to the right of the hedge cue exists a phrase introduced by 'because', 'since', 'like', 'unlike', 'unless','minus','although', 'i.e.', or 'as' (when preceded by a comma), exclude it from the constituent scope.

- If the constituent is a noun phrase, include in the scope every prepositional phrase to its right. For example 4.10 the scope of the parent noun phrase is extended to include the prepositional phrase 'of approximately 0.1 nM' (the sentence syntax tree is shown in

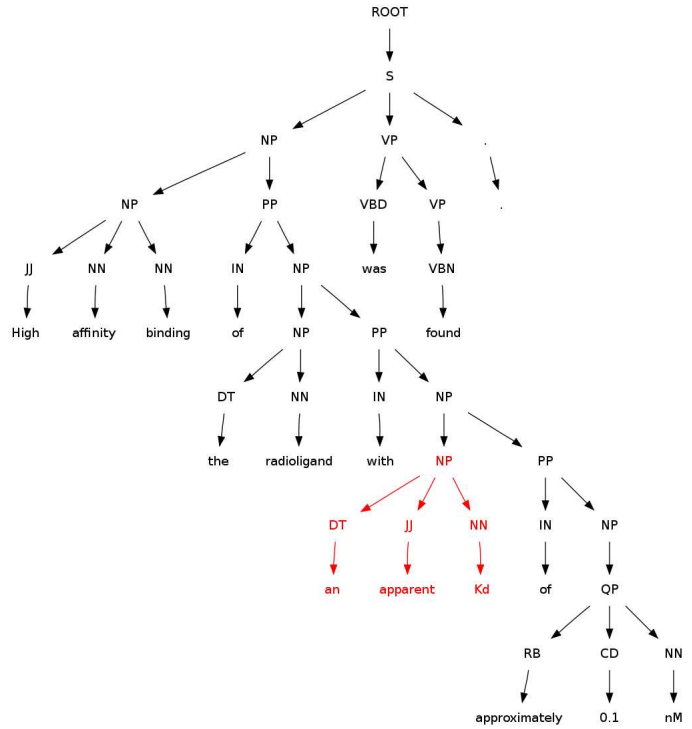

Figure 4.11: Syntax tree for sentence 4.10. The parent noun phrase of the hedge cue is shown in red

Figure 4.11). Note that this could also be considered a case of wrong ancestor selection, since including the PP to the right is the same as considering the enclosing NP as the hedge scope.

(4.10) High affinity binding of the radioligand with an {apparent Kd of approximately 0.1 nM} was found.

- Finally, if the constituent is a noun phrase and the hedge cue is an adjective, exclude (if it exists) the determiner to the left of the hedge cue. In the previous example, this rule excludes from the parent noun phrase scope the initial determiner 'an'. This seems a strange rule, and probably overfits the training corpus, but the error appeared in some cases in the held out corpus, so we decided to include a rule to cope with the situation, following the methodology.

Configuration	F	New errors	Solved errors
Baseline	0.664	-	-
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	0.705	5	11
Iteration #2	0.740	12	36
Iteration #3	**0.756**	7	15

Table 4.29: Classification performance on the held-out corpus for scope detection, after iteration #3.

After applying these adjustments, the match between hedge scopes and ancestor scopes improved from 80% of the hedge cues to 85% in the training corpus.

We expected this modification in ancestor scopes to improve classification, and this was indeed the case: F-score on the held-out corpus improved by two percentage points. Table 4.29 updates the evaluation results on the training corpus, and compares them with previous iterations.

4.6.5 Iteration 4: Handling Misclassified examples

In section 4.6.3 we identified a third source of errors beside wrong scope selection and mismatches between hedge and syntactic scopes: the cases where the classifier failed to classify one sentence token as the first element of the scope and one token as the last one. In these cases, we could be sure that this evaluation instances would be misclassified, because they were 'badly-formed', since they did not meet the very definition of scope. After the last iteration, 56 of the 96 classification errors corresponded to these cases.

As we have previously mentioned, these cases were handled using postprocessing rules, based mainly in positional and lexical information. Since we had available the sentence syntax structure, we aimed to use the annotation guidelines (strongly tied with this information) to improve classification. We therefore reviewed the remaining classification errors, and substituted all the postprocessing rules with a new set based on syntax:

- If the hedge cue is a conjugated verb (except when in passive voice or in the case of raising verbs such as 'seem'), use the next verb phrase up in the syntax tree that includes it.
- If the hedge cue is 'or', 'neither' or 'either', use the first noun phrase that includes it.

Configuration	F	New errors	Solved errors
Baseline	0.664	-	-
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	0.705	5	11
Iteration #2	0.740	12	36
Iteration #3	0.756	7	15
Iteration #4	**0.860**	2	42

Table 4.30: Classification performance on the held-out corpus for scope detection, after iteration #4.

- In every other case, use the first clause that includes the hedge cue.

After modifying postprocessing rules, performance dramatically improved by more than ten percentage points, as table 4.30 shows. The proposed heuristics solved 42 of the 56 errors, introducing only 2 new errors. To further investigate such improvement (the most important so far in our process), we studied how much each rule contributed to it. We found that, from the almost 10 improvement points, 5.9 were due to the third rule ('if you do not have cues, select the next enclosing clause'), 3.4 come from the first rule ('use the next VP for conjugated verbs'), while the improvement due to the 'or' rule was negligible. It seemed that rules based on syntax worked much better than the previous ones, which used mainly lexical information.

4.6.6 Iteration 5: Postprocessing Rules

After we had studied the three types of errors identified in section 4.6.3 and modified the classifier attributes and postprocessing rules, improving classification, we had only 49 errors left. We studied these errors, and identified several patterns where the classifier did not manage to predict the correct scopes. We found that in most cases these errors probably corresponded to situations where, despite having enough attributes, the absence of enough training data prevented the classifier to infer them. To try to solve this problem, we took a rule-based approach: we ignored the classifier predictions and deterministically assigned the scope limits. We had to be careful about being very precise in the determination of the situations these rules fired, the avoid introducing false positives. The rules we added are listed below:

- If the hedge cue is a verb, conjugated in passive voice, use as scope the first clause in the syntax tree that includes it. In sentence 4.11, for example, the scope is the clause

'NF-kappaB may be required for human CD34(+) bone marrow cell clonogenic function and survival.'

(4.11) In addition, we demonstrate that {NF-kappaB may be required for human CD34(+) bone marrow cell clonogenic function and survival}.

- If the word to the left of the predicted scope is 'which', include it in the predicted scope, as in sentence 4.12.

(4.12) To define the mechanism of action of the Nef protein, the signal transduction pathways which may be affected in T cells by constitutive expression of the nef gene were examined.

- If the hedge cue is 'likely', and it is preceded of followed by the verb 'to be', then the predicted scope is the first clause in the syntax tree that includes it. For sentence 4.13, the predicted scope matches the whole sentence, since it is the first clause that encloses the hedge cue.

(4.13) {The AP-1 site at bp, but not the NF-kappa B site, is likely to represent the major target of protein kinase C in the interleukin 2 promoter}.

- Finally, eliminate from every scope the bibliographic references at the end of the sentence (see, for example, sentence 4.14).

(4.14) {An ability of the Epstein-Barr virus latent membrane protein LMP1 to enhance the survival of infected B cells through upregulation of the bcl-2 oncogene was first suggested by experiments involving gene transfection and the selection of stable LMP1+ clones} (S.Henderson, M. Rowe, C.Gregory, F.Wang, E.Kieff, and A.Rickinson, Cell 65:1107-1115, 1991).

After these modifications we solved 10 classification errors, and introduced 3 new ones, yielding an improvement of F-score to 0.875. While we developed these rules, we felt they were too tailored to a very small number of errors, and wondered about its performance on unseen data. In the following chapter we will see that results confirmed our concerns: classifier performance actually decreased on evaluation data.

4.6.7 Tuning Learning Parameters

After we had found our 'best' classifier, we decided to tune the learning parameters, systematically trying different combination of attributes, with different windows sizes, and including

Configuration	F	New errors	Solved errors
Baseline	0.664	-	-
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	0.705	5	11
Iteration #2	0.740	12	36
Iteration #3	0.756	7	15
Iteration #4	0.860	2	42
Iteration #5	**0.878**	3	10

Table 4.31: Classification performance on the held-out corpus for scope detection, after iteration #5.

also bigrams and trigrams of input attributes. We also adjusted the CRF hyperparameter that traded the balance between overfitting and underfitting for best performance on the held-out corpus. The best configuration we found is shown in Table 4.32. Using this combination, we reached a maximal performance of 0.885 on the held-out corpus. Table 4.33 updates performance on the held-out corpus.

Since we based our improvements on classification errors, we risked overfitting the held-out corpus. To check if that actually happened, we evaluated the different classifiers on the previously unseen evaluation corpus. The following chapter shows the obtained results.

4.7 Summary

In this chapter we showed how the methodology presented in the previous chapter could be successfully applied to the sequential learning tasks of hedge cue identification and hedge cue detection. We tried all the time to follow the methodology guidelines, basing our analysis only on errors found on the held out corpus, to evaluate how far we could go just using that information. Results were promising: we ended with a classifier that, applied to the held out corpus, obtained more than 20 points of improvement in terms of F-score for scope detection (the most difficult of the two tasks). We also found the Knowledge Rules method a simple and clear way to suggest prediction rules and incorporate them to the learning analysis.

We had yet to know if the improvement hold on previously unseen data. In the next chapter we show that this was actually the case, and comment in detail the results and how different components of the learning architecture affected learning performance. We also discuss the importance of difference features for the selected tasks.

$q(y_{i-1}, y_i)$	$p(x, i)$
$y_i = y$ $y_i = y, y_{i-1} = y'$	true
	$hc_i = hc$
	$hc_{i-1} = hc$
	$hc_{i+1} = hc$
	$tp_i = tp$
	$tgp_i = tgp$
	$tp_i/tgp_i/tggp_i = m$
	$ps_i = ps$
	$ps_{i-1} = ps$
	$ps_{i+1} = ps$
	$ps_{i-2} = ps$
	$ps_{i+2} = ps$
	$gps_i = gps$
	$gps_{i-1} = gps$
	$gps_{i+1} = gps$
	$gps_{i-2} = gps$
	$gps_{i+2} = gps$
$y_i = y$	$t_i = t$
	$t_{i-1} = t$
	$t_{i+1} = t$
	$t_{i-2} = t$
	$t_{i+2} = t$
	$l_i = l$
	$l_{i-1} = l$
	$l_{i+1} = l$
	$l_{i-2} = l$
	$l_{i+2} = l$
	$es_i = es$
	$es_{i-1} = es$
	$es_{i+1} = es$
	$es_{i-2} = es$
	$es_{i+2} = es$

Table 4.32: Attributes for the final classifier. hc_i represents the BIO tag of the current word, l_i its lemma, t_i its POS tag, tp_i, tgp_i and $tggp_i$ are the POS tags of the ancestors of the hedge cue in the syntax tree, while ps_i, gps_i and $ggps_i$ are the attributes that mark their yields in the syntax tree. Attribute es_i represents the yield of the enclosing scope, calculated as in the postprocessing rules presented in 4.6.5

Configuration	F	New errors	Solved errors
Baseline	0.664	-	
Iteration #1 (without X-rule)	0.687	15	27
Iteration #1 (with X-rule)	0.705	5	11
Iteration #2	0.740	12	36
Iteration #3	0.756	7	15
Iteration #4	0.860	2	42
Iteration #5	0.878	3	10
Adjusted Parameters	**0.885**	6	10

Table 4.33: Classification performance on the held-out corpus for scope detection, after parameter tuning

5

Results

In this chapter we measure the performance of the different classifiers on the evaluation corpus, trying to predict performance on future data. We also show that recognition improvement achieved on the held out data during the development process holds for the evaluation corpus, and that some improvement results are statistically significant. We also compare the performance of our best classifier on a publicly available corpus, the CoNLL Shared Task 2010 corpus (Farkas et al., 2010b), with state-of-the-art methods, showing our results are competitive, and analyse qualitatively the final classification errors, trying to identify their causes and characterize the difficulties each task poses.

5.1 Performance on the Evaluation Corpus

As we have mentioned in section 4.3, the evaluation corpus we used comprises about 20% of the abstracts section of the Bioscope corpus, including 2302 sentences, 17% of them speculative, including 524 hedge cues. To evaluate performance, we rebuilt the classifiers in each one of the method's iteration, this time training on the whole training corpus (also including sentences in the held out corpus). We do not only wish to estimate the prediction ability of our best classifier on future data, but also to evaluate if the application of the methodology actually produced a performance improvement after each iteration, when faced with unseen data.

Results are measured in terms of the usual figures of precision, recall and F-score, as we have previously done on the held out corpus. We additionally included the F-score for the different non-O tags in each task (i.e. `B` and `I` for hedge cue identification and **F** and **L** for scope detection), trying to investigate how they affected the classifier performance. The next sections show results for the two tasks with additional figures seeking to evaluate how certain devices (X-Rules and postprocessing rules) influenced on the final results. Section 5.1.3 present the

Classifier	Precision	Recall	F-score	B F-Score	I F-Score
Baseline	**0.979 (0.955)**	0.781 (0.740)	0.868 (0.834)	0.871	**0.806**
Initial Classifier	0.963 (0.944)	0.847 (0.805)	0.902 (0.869)	0.907	0.795
Improved Classifier	0.947 (0.913)	**0.866 (0.840)**	**0.915 (0.875)**	**0.922**	**0.806**

Table 5.1: Hedge cue recognition: results on evaluation data. Number in parenthesis show results on the held out corpus. The last two columns show the F-measure for the task of identification of `B` and `I` tags

results of the application of the two classifiers in sequence, showing how hedge cue recognition errors affect the final system results.

5.1.1 Hedge Cue Identification

Table 5.1 shows the performance of the classifier built after each iteration for the task of hedge cue recognition (parenthesized figures recall the results on the held out corpus for comparison purposes).

We can see that classifier improvement on the held out corpus holds in the evaluation corpus. For the hedge cue recognition task, both precision and recall increased compared with the held-out corpus results, suggesting that the availability of more training data allowed to improve results (this improvement seems limited, as section 5.3 further discuss).

Scores for each tag identification are higher than the final F-score (this is trivially true, because to correctly identify the complete hedge we must correctly tag each word it includes). However, we can observe that, while identify the `I` tag seems more a difficult task (and the attributes here suggested could not actually improve it), the influence of the identification of the first token of the hedge cue is much higher (the final F-Score is less than one point in every case), probably due to the preeminence of one-word hedge cues.

As we mentioned in section 4.4, during the development of the classifiers we always used Markov order 1 CRFs (that is, the feature functions we used for learning depended on the current and previous output tags). We aimed to know, after we built our best classifier, how much taking into account the previous tag influenced in the classifier performance. When we trained a new classifier using Markov order 0 CRFs, F-score dropped only about half a point (from 0.916 to 0.911), suggesting that, for this task, the use of a per-token classification method could be enough to achieve top performance.

Classifier	F-score	F F-score	L F-score
Baseline	0.737 (0.664)	0.865	0.839
It. #1 (Adding syntax info., without X-rule)	0.752 (0.687)	0.879	0.883
It. #1 (Adding syntax info.,with X-rule)	0.749 (0.705)	0.877	0.874
It. #2 (Adding ancestors)	0.802 (0.740)	0.887	0.900
It. #3 (Adjusting ancestors scopes)	0.800 (0.756)	0.882	0.909
It. #4 (Handling misclassified examples)	**0.852** (0.860)	**0.932**	**0.916**
It. #5 (Postprocessing rules)	0.837 (0.878)	0.914	0.916
Adjusted parameters	0.831 (**0.885**)	0.911	0.908

Table 5.2: Scope detection: results on evaluation data. Number in parenthesis show results on the held out corpus.The last two columns show the F-measure for the task of identification of `F` and `L` tags, i.e. the identification of left and right scope limits

5.1.2 Scope Detection

Table 5.2 shows the performance of the classifier built after each of the the eight iterations for scope detection (in this case, we only report F-score since, as we explained in the previous chapter, the problem formulation implies precision equals recall). Parenthesized figures recall the results on the held out corpus for comparison purposes.

For this task, again improvement on the held out corpus holds in the evaluation corpus. We can observe that for the first classifiers, results in the evaluation corpus where better than those for the held out corpus, while, after iteration #4, results were worse. This could suggest that the method of using errors on the held out corpus to drive classifier improvement produced a certain level of overfitting to the held out corpus, producing anyway better classifiers. We can also note that for the two last classifiers (which incorporate hand-made rules and parameter tuning) performance actually *decreases*, suggesting respectively that these rules were too specifically tailored to the held out corpus and that this held out corpus could actually be too small to use it for tuning (an alternative explanation appears when we we consider performance before postprocessing wrongly predicted scopes, as is later shown).

The observation that the best results for left and right scope limit identification coincided with the best scope detection results show that classifier improvement was a consequence of better boundary identification (even when, for several cases, it did not implied correct scope detection, as the difference between final scores and boundary identification scores show).

An unexpected result was that, for iteration #1, adding the X-rule actually hurt scope detection accuracy. To better analyze the result, we measured how the use of X-rules impacted

Classifier	F-score before postproc.	F-score after prostproc.
Baseline	0.552	0.737
It. #1 (Adding syntax info., without X-rule)	0.582	0.752
It. #1 (Adding syntax info.,with X-rule)	0.578	0.749
It. #2 (Adding ancestors)	0.637	0.802
It. #3 (Adjusting ancestors scopes)	0.635	0.800
It. #4 (Handling misclassified examples)	0.635	**0.852**
It. #5 (Postprocessing rules)	0.635	0.837
Adjusted parameters	**0.687**	0.831

Table 5.3: Impact of postprocessing rules on scope identification

the performance of our best classifier. When we eliminated X-Rules (predicting only with per-token attributes), performance degraded from 0.839 to 0.835, a difference that seems too small to let us extract clear conclusions. It seems that X-Rules (at least those used in our experiments) reduced their impact on classifier performance when more training data were available. Future research should be done to clarify this point.

We also wanted to evaluate the performance of using postprocessing rules to handle those cases where the predicted scopes were badly formed (i.e. did not included exactly one token as left scope boundary and one token as right scope boundary). To do this, we compared, for each classifier, the scope detection accuracy before and after postprocessing rules. Results are shown in table 5.3.

The first observation is that postprocessing rules are actually *very* important for the task. About 20 points of F-score were due to cases correctly predicted by these rules. This also explains the important improvement after including rules to use ancestor scopes for correcting inconsistent scopes. The other clear source of improvement (the incorporation of ancestor scopes as learning attributes) is, however, explained by the better performance of the original classifier. This indicate that both the supervised learning method *and* the handwritten rules impact the final classifier performance.

Another thing to note is that parameter adjusting actually worked very well on improving the original classifier performance, but it seems that the new correctly predicted cases were already solved by the postprocessing rules, while some previously correctly predicted cases were lost, and they could not be solved with the rules, yielding an overall performance decrease.

We repeated the experiment we did with hedge cue identification, and trained a new classifier using Markov order 0 CRFs with the same attributes than our best classifier. In this case,

Classifier	F-score (Golden H.Cue)	F-score (Guessed H. Cue)
Baseline	0.737	0.704
It. #1 (Adding syntax info., without X-rule)	0.752	0.713
It. #1 (Adding syntax info.,with X-rule)	0.749	0.707
It. #2 (Adding ancestors)	0.802	0.754
It. #3 (Adjusting ancestors scopes)	0.800	0.731
It. #4 (Handling misclassified examples)	**0.852**	**0.785**
It. #5 (Postprocessing rules)	0.837	0.774
Adjusted parameters	0.831	0.769

Table 5.4: Comparison of hedge cue identification and scope detection results when using golden hedge cues and guessed hedge cues

F-score dropped almost two points (from 0.852 to 0.835).

5.1.3 Overall System Performance

During the classifier development process, we assumed for the scope detection task that hedge cues had already been correctly identified (i.e., used the gold standard hedge cues), as we previously mentioned at the end of section 4.3. To measure the effectiveness of the system in real life situations, where we start only with the input sentence and want to identify hedge cues *and* their scopes, we should consider results after applying both classifiers in sequence, measuring how hedge cue identification impacts on the final results. To do this, we followed the approach of the CoNLL Shared Task, and considered a hedge cue correctly identified only if both the hedge cue and the scope were correctly predicted. This could be seen too restrictive, since it could be possible that a multiword hedge cue could be partially recognized and yet the scope correctly identified, but it seems a clear measure for comparative purposes. Table 5.4 compares performance of the scope detection classifiers using gold standard versus predicted hedge cues, showing that, as it could be expected, performance decreases and the relative results of the different classifiers remain the same.

5.2 Cross Validation

A question remains: it is possible that, even when the classifier is rebuilt on the new training corpus, the *method* used was too tailored to the corpus we used for improving it? This seems

Classifier	Average F-Score	Confidence interval
Baseline	0.853	[0.849,0.857]
Initial Classifier	0.882	[0.875,0.889]
Improved Classifier	**0.886**	[0.880,0.893]

Table 5.5: Hedge cue recognition: results after cross validation.

Classifier	F-score	Confidence interval
Baseline	0.712	[0.699,0.725]
Iteration #1 (without X-rule)	0.729	[0.718,0.740]
Iteration #1 (with X-rule)	0.734	[0.720,0.748]
Iteration #2	0.765	[0.755,0.774]
Iteration #3	0.772	[0.755,0.789]
Iteration #4	**0.835**	[0.815,0.855]
Iteration #5	0.828	[0.814,0.841]
Adjusted parameters	0.826	[0.878,0.894]

Table 5.6: Scope detection: results after cross validation.

reasonable, since we based our decisions on committed errors rather than on general corpus analysis. To see whether that actually happened, we performed a ten fold cross-validation, splitting the whole learning corpus into ten parts, training each classifier on nine folds and evaluating on the tenth, repeating the process changing the evaluation fold and averaging results, to reduce their statistical variance. To split the corpus we randomly assigned every document to one of the ten folds.

Tables 5.5 and 5.6 show the average F-score and a 95%-confidence interval of each classifier, considering the different F-scores as normally distributed for both tasks. In both cases, we found that (except for the two last classifiers for scope detection) improving performance on the held out corpus implied improving on the whole corpus.

We have yet to know if this improvement was statistically significant. To measure this, we performed a Wald test (Wasserman, 2003) to determine if the difference between mean F-scores could be due to chance, with 95% confidence. Given two F-scores (corresponding to the average evaluation result for two classifiers), the null hypothesis we want to reject is that the difference between the two means is zero. Following the Wald test method, we specify that

Classifier	Average F-Score	W-value
Baseline	0.853	
Initial Classifier	0.882	**6.89**
Improved Classifier	0.886	0.80

Table 5.7: Hedge cue recognition: statistical significance of each improvement

Classifier	F-score	W-value
Baseline	0.712	
Iteration #1 (without X-rule)	0.729	1.86
Iteration #1 (with X-rule)	0.734	0.50
Iteration #2	0.765	**3.34**
Iteration #3	0.772	0.72
Iteration #4	0.835	**4.48**
Iteration #5	0.828	-0.58
Adjusted parameters	0.826	-0.13

Table 5.8: Scope detection: statistical significance of each improvement.

$$W = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_1^2}{10} + \frac{s_2^2}{10}}}$$

where $\overline{X}$ and $\overline{Y}$ represent the scores average values, while s_1^2 and s_2^2 are the scores variances. We reject the null hypothesis (i.e. suppose that improvement is not by chance) if $|W| > 1.96$, the limit of a 95% probability interval for a standard Normal distribution. Table 5.7 shows the results for hedge cue identification, while Table 5.8 does the same for scope recognition.

Results for hedge cue identification show that the initial classifier (obtained tuning window size of the different learning attributes) produced a statistically significant improvement, while we cannot conclude that adding cooccurrences actually will improve results on future data.

For the scope detection task, two improvement methods yielded significant results: adding syntax information to the learning attributes (through knowledge rules), and postprocessing results using syntactic information to correct invalid or incomplete scopes. The remaining improvements were not found to be statistically significant, meaning we should evaluate them on more data to asses its effectiveness.

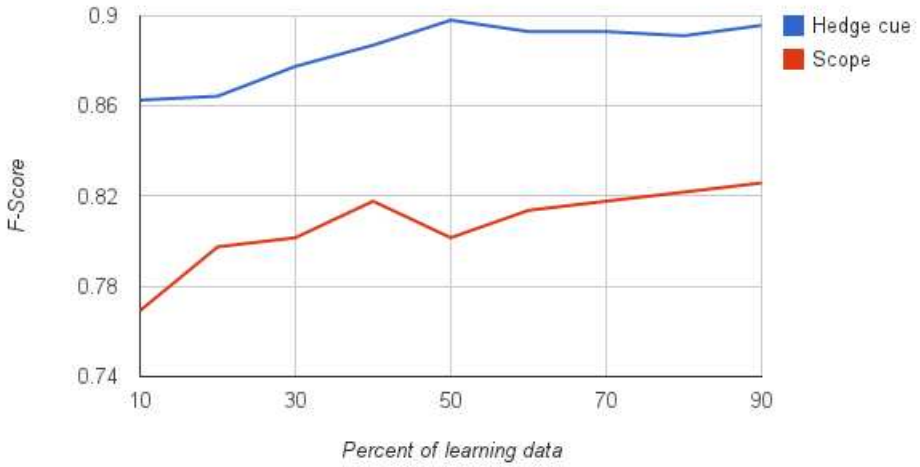

Figure 5.1: Learning curves for hedge cue identification (shown in blue) and scope recognition (shown in red)

5.3 Learning Curve

Trying to characterize the posed problems, we evaluated how much the training information available impacts on classifier performance for each task. We trained on a certain proportion of the corpus, and evaluated the classifier on one of the folds built in the cross validation process. Figure 5.1 shows the learning curve for both tasks, showing how F-score varied with the percentage of the learning corpus used. Table 5.9 shows the obtained scores.

We can see that after processing half of the corpus, the learning performance of the classifier for hedge cue identification stopped increasing, and even decreased. This suggests that the number of different hedge cues used (at least in abstracts) is limited, and also that each hedge cue appears in similar positions in the sentence, causing that the amount of training data needed to achieve peak performance is not too large. A question remains: why, if it is so easy, we still get about 90% F-score? If we observe the performance when training on the whole training set (including evaluation data during training), we found that performance raises to almost 100%, showing that there are almost no 'contradictory' training instances (that is, instances with the same attribute values and different target classes). We can conclude that the difference come from hedge cues *not appearing* or appearing very few times in the training corpus: it seems

% of the corpus	F-score (HC)	F-score (Scope)
10	0.863	0.769
20	0.864	0.798
30	0.878	0.802
40	0.887	0.818
50	**0.898**	0.802
60	0.893	0.814
70	0.893	0.818
80	0.891	0.822
90	0.896	**0.826**
100	0.996	0.907

Table 5.9: Learning curve for hedge cue identification (HC) and scope recognition

that is very difficult to learn to identify new hedge cues. In the following section we perform a qualitative analysis that seems to confirm this thesis.

The learning rate for the scope recognition task is slightly different. An obvious observation is that scope recognition results are consistently lower than those for hedge cue identification, no matter how much data we use for training, showing that the second problem is harder than the first. Referring to the learning rate, it seems that more information could imply better results for the task. However, if we look at the numbers when we train on the whole learning set, the classifier achieved a 90% F-score, showing that, even overfitting the training set, there is still an important number of misclassified instances: this indicates that there are different target classes assigned to similar features. This could in turn be due to annotation inconsistencies, or because the model is not rich enough to separate those cases.

5.4 Error Analysis

In this section, we study the errors committed by the classifiers, trying to explain why they occurred. We first show an analysis of misclassified hedge cues, suggesting that hedge cue identification is very sensitive to the number of occurrences of each hedge cue in the training corpus. We then analyze scope recognition errors and group them using their possible error causes.

Hedge cue	TP	FN	FP	HC in TC	Total app. in TC
unknown	2	9	0	14	38
or	0	8	1	77	928
could	9	5	2	51	128
potential	8	5	0	32	114
indicate	1	3	1	4	199
not known	2	3	0	4	6
either or	0	3	0	24	149
can not be excluded	0	2	0	1	1
potentially	4	2	0	13	17
proposed	1	1	1	3	7
possibility	2	1	1	6	18
hypothesis	6	1	1	11	19
elusive	0	1	0	0	1
remains to be elucidated	0	1	0	0	1
can not be	0	1	0	0	12
suspect	0	1	0	0	2
yet to be understood	0	1	0	0	0
not been clearly elucidated	0	1	0	0	0
uncertain	0	1	0	0	1
hypothesised	0	1	0	0	0
must	0	1	0	0	8
not clearly delineated	0	1	0	0	0
not fully understood	0	1	0	0	3
hypothesize	1	1	0	2	9
not clear	0	1	0	4	4

Table 5.10: Instances not classified as hedge cues, including number of occurrences as hedge cues in the training corpus and total number of appearances. TP stands for True Positives (instances correctly classified), FN for False Negatives (instances not classified as hedge cues when they should), FP for False Positives (instances classified as hedge cues when the should not)

5.4.1 Hedge Cue Identification

Table 5.10 shows the complete list of hedge cues the classifier failed to identify in the evaluation corpus, sorted by the number of times this happened. For each case, we also show the number of times they appeared in the training corpus as hedge cues and their total number of occurrences there. We can see they include hedges such as 'or','could' and 'potential', whose proportion of times appearing as hedge cues is small compared with their total number of occurrences. Some words such as 'hypothesis' or 'potentially' appear in the training corpus a similar number of times acting as hedge cues and not doing so, causing the classifier to sometimes mark them as hedge cues, while failing in other cases. The most common source of errors is the hedge cue 'unknown': the classifier fails 9 of 11 times in recognizing it as a hedge cue. Analyzing the training corpus, we found that is very difficult to determine, even for a human being, if it acts as a hedge cue. For example, in the following sentence:

(5.1) Sublethal levels of oxidative stress are well known to alter T cell functional responses, but the underlying mechanisms are unknown.

the word is not marked as a hedge cue, while in the sentence:

(5.2) The mechanisms by which beta-catenin undergoes this shift in location and participates in activation of gene transcription are unknown.

annotators have considered a speculation cue. Similar examples can be found for the word 'or'. Since annotation criteria for the corpus are not stated clearly, we cannot determine if this was an annotation decision, or simply an annotation error.

This list also includes eleven cases of hedge cues that were not present in the training corpus. Conversely, looking at the complete list of errors, we found that there was no case of hedge cue that did not appear in the training corpus and were correctly classified, confirming that identifying new hedge cues only from its context is a very difficult problem.

False Positives (i.e. instances incorrectly marked as hedge cues) are less common than False Negatives (26 cases against 56), and is more difficult to guess why the classifier was wrong. Some of them ('indicate specific', 'indicate involvement') are extensions of a common hedge cue ('indicate'): the classifier decided to include the following token within the hedge cue. In the remaining cases, it was probably the context which misled the classifier. The only case of 'indicate that' that was not correctly classified, probably corresponds to an annotation omission:

(5.3) Transfection studies of LTR reporter constructs indicated that mutation of the DSE sites abrogated the LTR-mediated synergy induced by Ctx and TNFalpha,...

Table 5.11 shows the complete list of tokens incorrectly classified as hedge cues.

Hedge cue	TP	FN	FP	HC in TC	Total app. in TC
considered	0	0	3	3	11
could	9	5	2	51	128
apparently	2	0	2	5	14
putative	3	0	2	34	48
or	0	8	1	77	928
indicate	1	3	1	4	199
proposed	1	1	1	3	7
possibility	2	1	1	6	18
hypothesis	6	1	1	11	19
indicate specific	0	0	1	0	0
either	0	0	1	0	143
indicate involvement	0	0	1	0	0
predicts	0	0	1	1	4
conclusion	0	0	1	1	12
assumed	0	0	1	2	3
apparent	0	0	1	5	36
perhaps	0	0	1	7	7
indicating	1	0	1	11	63
propose	7	0	1	25	39
indicated that	15	0	1	40	41
can	0	0	1	43	356

Table 5.11: Instances wrongly classified as hedge cues

Error type	Number of cases	%
Parser Errors	19	25
Modified Scope Errors	11	14
Scope Selection Errors	17	22
Passive Voice Annotation Errors	5	6
Noun, Adjective, and Adverb Annotation Idiosyncrasies	10	13
Other Errors	15	20

Table 5.12: Classification Error Categorization

5.4.2 Scope Recognition

We studied the 77 errors committed by the scope detection classifier on the evaluation corpus, trying to identify their causes. Table A.1 in appendix A shows the complete categorized list of errors. In this section we group them according to the possible error cause, and show some examples for each case. Table 5.12 summarizes these results, including the number of cases each group comprised.

Parser Errors

The most common error cause corresponded to parser errors (about 25% of the total number of errors). This should be no surprise, since the sentence syntactic analysis was not done by hand, but using an external constituent parser. A common parsing error, that of wrong phrase attachment, modified the syntactic scope of the ancestors of the hedge cue, leading to cases where the classifier selected the correct ancestor, but with the wrong scope. This situation was identified during the improvement phase (refer to section 4.6.4) and we tried to solve it including scope modification rules (since we could not modify parser results). Still, results show that some errors could not be solved.

Consider, for example, the following sentence, where the gold standard scope is marked within curly braces, while the predicted scope is marked within brackets.

(5.4) This activation is inhibited {[by known inhibitors of NF-kappa B or by simultaneous treatment of the cells] with surfactant lipids}.

In this case, the parser incorrectly attaches the PP started with the word ‘with’ to the VP headed by ‘inhibited’, instead of attaching it to the NP, reducing the scope of the coordination. The classifier correctly selects as scope the syntactic scope of the coordination, but, since it does not include the PP, the result is incorrect.

Modified Scope Errors

The rules mentioned in the previous section were thought to correct common parsing errors and reflect some annotation idiosyncrasies, and results show that they were successful. However, there were some cases where they incorrectly pruned the syntactic scope, leading to analysis errors. In the following example, a scope-pruning rule incorrectly excludes from the hedge scope the clause introduced by 'although':

(5.5) Further Northern analyses {[indicated that there was no significant change in 17beta-HSD IV or DNA-PK(CS) mRNA levels following treatment with 1,25(OH)2D3], although expression of both genes varied with changes in cell proliferation}.

This case also shows that, in some cases, is not clear which should be the correct scope, according to the annotation guidelines; it looks like the decision of whether the clause from the previous example should be included depends more on semantic observation than on syntax. We will come back to this issue on the following chapter.

These type of errors accounted for about 14% of the total number, and are identified in appendix A. The reader is referred to that table for more modified scope examples.

Scope Selection Errors

Most errors (about 78% of the total) correspond to cases where hedge scope did not coincide with the scope of any ancestor in the syntax tree (considering the modified scope definition presented in section 4.6.4). The classifier seemed very good at finding the correlation between the correct ancestor scope and the final hedge scope, when that was possible. However, there were some cases where it failed, and selected the wrong ancestor. For example, in the following sentence, the classifier selects the syntactic scope of the parent ancestor in the syntax tree as the hedge scope, while the gold annotation marked the grandparent scope.

(5.6) {Nuclear factor-kappa B (NF-kappa B)/Rel transcription factors [may be involved in atherosclerosis], as is suggested by the presence of activated NF-kappa B in human atherosclerotic lesions}.

The most probable explanation for these type of errors is that the classifier learns to identify the scope with certain ancestor (in the example, the parent), because that is the most common situation in training data for certain attribute values such as, for example, the hedge cue tokens, missing the cases where, due to the particular sentence structure, the scope coincides with another ancestor. Even when we included the POS-tag of the syntactic constituent to help with these cases, it seems that it was not enough.

Passive Voice Annotation Errors

We found five cases (6% of the classification errors) where it seems that the original annotation was wrong, because they mark the verb phrase instead of the full clause in the case of passive voice, something that contradicts the corpus annotation guidelines. The following example shows one of those cases:

(5.7) We therefore propose that [such cross-talking among distinct adhesion molecules {may be involved in the pathogenesis of inflammation, including RA synovitis}].

Noun, Adjective, and Adverb Annotation Idiosyncrasies

About 13% of the classification errors corresponded to problems with the scope of noun, adjective and adverb hedge cues. How these scopes should be annotated is not clear in the corpus annotation guidelines. In the case of the scope of attributive adjectives, for example, the guidelines state that 'generally extends to the following noun phrase', while 'the scope of predicative adjectives includes the whole sentence'. In the first case, it looks like other premodifiers of the noun phrase should not be included within the scope: in this case, the scope does not correspond to a syntax constituent (against what is stated in the general annotation guidelines). Since our classifier (due to the attributes used) tends to identify scopes with syntactic constituents, selected scope often does not match the gold standard annotation. A similar situation happens with nouns (we generally select the noun phrase including the hedge cue) and adverbs. The following examples show some of these cases:

(5.8) Findings that [xenogeneic serum promotes leukocyte-endothelium interaction {possibly through NF-kappa B activation}] might be relevant for designing future therapeutic strategies aimed at prolonging xenograft survival.

(5.9) Our results suggest that G0S2 expression is required to commit cells to enter the G1 phase of the cell cycle, and that, while not excluding [other {possible targets}], early inhibition of G0S2 expression by CsA may be important in achieving immunosuppression.

(5.10) [An alternative {possibility is that in the absence of gamma(c), the IL-4R alpha chain is able to transduce signals by homodimerization}].

Other Errors

The remaining errors correspond to several, more subtle situations: there are some cases where some of the attachments proposed by the parser are (even when probably correct) not the same

as those suggested by the gold standard; there are also situations where the parser included item numbers, bibliographic references or punctuation marks within the scope, yielding slightly different scopes. We included several rules to cope with these cases if they appeared during the improvement phase of the methodology, but they were not enough to manage every possible situation.

5.5 Evaluation on the CoNLL 2010 Shared Task Corpus

The main aim of our work was to evaluate how performance improved after adding expert knowledge, and that is what the results in the previous sections show. However, we thought it could be useful to compare our results with state-of-the-art methods, trying to determine how far we could go using the proposed methodology. To achieve this, we used the corpus of the CoNLL 2010 Shared Task (Farkas et al., 2010b) to train and evaluate our classifiers, using the configurations that had achieved top performance on the evaluation corpus.

Recall from section 2.2.5 that the Shared Task corpus included the abstracts section from the Genia corpus (i.e. the same set of documents we used for our experiments), plus five full articles from the functional genomics literature and four articles from the Bioinformatics website, for a total of 14,541 sentences, annotated with the same guidelines used for the original Bioscope corpus. We can say that the CoNLL corpus is an augmented version of our training corpus, being probably the main difference the source for the added sentences, exacted from full papers rather than from abstracts. The evaluation data set, in turn, was based on 15 biomedical articles from the PubMed databases, containing 5003 sentences, out of which 790 where uncertain. In this case, the complete corpus came from full articles, something that had a negative effect on performance, as several Task teams have reported (Farkas et al., 2010b; Morante et al., 2010). Besides comparing our method with state-of-the-art procedures, the use of this corpus allows us to evaluate how our classifiers behave when faced with a slightly different

	Hedge cue identification			Scope detection		
	Prec.	Recall	F-Score	Prec.	Recall	F-Score
Baseline Classifier	**0.904**	0.645	0.753	0.428	0.397	0.412
Best Classifier	0.832	0.768	0.799	0.567	0.528	0.547
Best Results	0.817	**0.810**	**0.813**	**0.596**	**0.552**	**0.573**

Table 5.13: Classification performance compared with best results in CoNLL Shared Task. Figures represent Precision/Recall/F-score

type of texts (those including full articles) within the same domain, and annotated based on the same guidelines.

Table 5.13 shows the evaluation results. For the scope detection task, measures were obtained using the hedge cues learned by the hedge cue classifier, so the numbers represent an evaluation of the combined action of the two classifiers. The first thing to note is that results are clearly lower than those obtained in the evaluation corpus, even when the training corpus was larger. This was a common situation in the Shared Task (results were consistently lower than those obtained in studies produced before the Task), and is probably due to the fact that the full text articles used for evaluation included new hedging cues and shallow text features (a common example is the use of references at the end of sentences).

Our best classifiers obtained competitive results in both subtasks. For hedge cue detection, our best classifier achieved an F-measure of 0.799, better than the third position in the Shared Task for hedge identification, and more than four points over the baseline classifier. Scope detection results (using learned hedge cues, and so propagating hedge cue identification errors) achieved an F-measure of 0.547, performing better than the fifth result in the corresponding task, five points below the best results obtained so far in the corpus (Velldal et al., 2012), and 16 points better than our baseline system.

5.6 Conclusion

All the results presented in this chapter suggest that the methodology proposed in chapter 3 was successful, producing competitive classifiers and clearly improving baseline results to both tasks addressed. While some degree of overfitting could have existed, the final classifiers were clearly better on unseen data on the same corpus. When we trained an evaluated them on an augmented version of the corpus (based on the same annotation guidelines but with some new idiosyncrasies, because it included full text articles besides abstracts), performance improvement hold, and results were comparable to state-of-the-art figures.

An open question is how portable are these classifiers to other, different, corpus. Since they are built using hybrid methods, we are not very optimistic. However, the methodology has the advantage of clearly separating the data-driven from the rule-based aspects, so it is possible that the adaptation effort could be reduced to modifying the knowledge-intensive parts. Further research should clarify this.

6

Conclusions

This thesis studied the use of sequential supervised learning methods on two tasks related to the detection of hedging in scientific articles: those of hedge cue identification and hedge cue scope detection. Both tasks were addressed using a learning methodology that proposes the use of an iterative, error-based approach to improve classification performance, obtaining competitive results. The methodology assumes that the problem can be stated using a supervised classification approach, and suggests ways to improve the classifier by analysing classification errors on a held out corpus, with the help of a domain expert, and incorporating manual classification rules into the learning process. In this chapter we summarize the different steps we took to address the tasks, comment on the main difficulties addressed, and draw conclusions with respect to the methodology and its application to the above-mentioned tasks. Finally, we present some possible future lines of research, based on these conclusions.

6.1 Process Summary

Before delving into the tasks, we first studied the literature about hedging and the related theory of modality, trying to characterize the problem. To evaluate our work, we selected a publicly available dataset in the biological research domain, the Bioscope corpus, and enriched it by adding information resulting from lexical and syntactic analysis (using available external tools). We then considered the case of hedge cue identification and scope detection as special cases of sequential classification, following previous approaches in the literature: the hedge identification task consisted in determining whether each token in the sentences was part of a hedge cue, while for the scope detection task we tried to identify, given a sentence and a hedge cue, the first and final token of the scope induced by the hedge cue.

Once we had the corpus and the tasks clearly defined, we defined the methodology to use,

which was presented in chapter 3. Once we had determined an initial set of attributes, and had built a sequential classifier based on Conditional Random Fields, we analysed the errors committed and iteratively added new attributes, trying to improve classification performance, and evaluating this improvement on a held out corpus. The process was repeated until no further improvement could be achieved. Two different kind of attributes were added after each iteration: some of them were simply new lexical or syntactic features extracted from the corpus, while others took the form of what we have called knowledge rules. In the second case, each attribute corresponded to the result of the application of a hand made rule that proposed a tentative classification of the tokens in the sentence, corresponding to the target class to which we aimed to assign them. These suggested classes were not actually assigned to the tokens as a definitive result, but instead added as new attributes to the statistical learning process, letting the classifier determine if they were correlated with the real target class in the whole corpus, and therefore relevant for the desired task

Results suggest that the approach was successful: for the hedge cue identification task, we obtained an improvement on the evaluation corpus of 2.5 points in terms of F-Score, compared with the baseline initial classifier; for the scope recognition task we obtained a dramatic improvement of almost twelve points of F-Score, after several iterations. We also found that these results were statistically significant for both tasks.

6.2 Methodology Remarks

Results for the two tasks we addressed suggest that the presented methodology could be applied to other classification tasks where the learning scenario presented in section 3.1 holds; i.e., we have a supervised classification task and a human expert available, who can examine committed errors and suggest rules to improve performance.

Based only on the analysis of errors and previous linguistic knowledge, we obtained competitive results for the two tasks we addressed. We could identify the relevant features and difficulties posed by the tasks, arriving to similar conclusions to alternative methods presented in the literature. If we consider, for example, the recent exhaustive analysis of Velldal et al. (2012), the problems there identified were similar to those we found: the (limited) association between syntactic and hedge cue scopes, the need to select the correct ancestor in the syntax tree (we proposed for this the use of competing learning attributes, while in the mentioned article they used a reranking procedure to select the best parse from a list), and the problem of the correct identification of scope boundaries.

Results on the evaluation data also showed that the methodology seems robust to overfitting: improvement on the held-out corpus hold on the evaluation corpus. However, the fact

that classifiers based on specific rules actually did not work so well in the final evaluation, suggest that we should try to build (after error analysis) rules general enough, avoiding rules specifically tailored to certain uncommon errors.

The idea of knowledge rules as learning attributes that reflect the expert predictions for certain instances) seemed useful. The classifier showed very effective in selecting the cases in which each rule should be applied, letting us to combine competing approaches (such as the different ancestor scopes) to improve performance. This allow us to incorporate different classification suggestions into the learning process, and seems a promising method for combining rule-based and machine learning methods.

The use of X-Rules to adjust scopes once the first word was detected was not so relevant as our first results suggested. Even when they actually produced a performance improvement, results are not conclusive, since it looks that (in our application) enough training data compensated their prediction ability. Still, it would be interesting to further apply similar rules in other learning cases to better evaluate their performance.

During the application of the different steps of the methodology, we maintained an efficiently accessible data structure to hold the corpus information, and we kept all the learning information in a relational data structure. Both approaches proved useful: they allowed us to develop analysis tools to evaluate the classification result on each evaluation instance, facilitating their study. They also allowed us to avoid regenerating every attribute in each iteration, thus limiting the computational effort required to calculate attributes not previously obtained in earlier iterations. This method can be used in other, different, classification tasks, instead of the traditional approach of generating all the information for learning each time it is needed.

6.3 Evaluation of the Hedging Tasks

6.3.1 Hedge Cue Identification

Results for hedge cue identification seem to indicate that the list of hedge cues we used in the abstracts sections of research articles (the domain we evaluated our classifier on) form a rather closed set: about 98% of the hedge cues in the evaluation corpus already appeared in the training corpus. This implies that solving the ambiguity of words or expressions related to their role as hedge cues is enough to achieve very good classification performance. We showed that disambiguation can be achieved using only surface, lexical and contextual information. On the other hand, the problem posed by the few new hedge cues appears to be a very difficult one: none of these hedge cues could be correctly classified by our best classifier. It seems that an exhaustive list of terms potentially useful as hedge cues may be needed to improve performance.

The use of cooccurrence information (i.e. marking those cases where two or more hedge cues appeared in the same sentence) seemed to improve classification performance, though the results cannot be considered statistically significant. Further studies on the topic could clarify this fact.

6.3.2 Scope Detection

The scope detection task appears to be a more difficult one: results were consistently lower than those for hedge cue identification. The main source of improvement for the task was the incorporation of syntax information: relating hedge cue scopes with the syntactic scope of the ancestors of the hedge cue proved to be a very good approach, while using the same information into postprocessing rules to correct those cases where the statistical classifier failed to predict complete scopes made it possible to solve several classification errors without introducing new ones.

However, there were still several instances that could not be correctly classified, especially those that did not correspond with the scope of any ancestor of the hedge cue. We found that the classification method used was highly successful in finding the correlation between syntactic scopes and hedge scopes, when that was possible. In almost every case the correct ancestor in the tree was selected by the classifier to be used as the hedge scope, independently of the hedge cue part-of-speech tag. Adjusting the original syntactic scope of the hedge cue ancestor to exclude certain spans (such as certain prepositional phrases or clauses at the end or at the beginning of the scope) proved to be useful to improve classification. Yet, the absence of clear annotation rules for the selected corpus, indicating when they should be included or excluded acted as a source of ambiguity, and we found several corpus instances where the reasons for pruning or not pruning the hedge scope were not clear. The rules for scope pruning that we developed were based purely on data observation, probably causing an overfit to the training corpus.

6.3.3 Comparison with Previous Work

The evaluation on a slightly different corpus in the same domain (that of the CoNLL 2010 Shared Task) allowed us to compare our results with those of previous work. They were competitive for both tasks. We obtained the third best known result for the hedge cue identification task (two F-score points below the best result), and the fifth for the scope detection task (less than five F-score points below the best know result), when trained on the same data.

6.4 Comments on the Corpus Annotation

The task we aimed to addresses was mainly a computational one: we wished to evaluate a learning methodology to predict hedges and scopes, learning from the Bioscope corpus. From this perspective, what we considered 'correct', for evaluation purposes, was the corpus annotation. However, during the learning process and after exhaustively examining classification errors, we observed what we consider certain problems in the corpus annotations, beyond annotation errors. We mention them as a contribution to the better characterization of speculative detection in Natural Language Processing.

First, recall from chapter 2 our definition (based on the definition of the Cambridge English Grammar for negation scopes) about hedging scopes as the part of meaning in the sentences that is hedged, clearly a semantic concept, albeit strongly related with syntax. The corpus annotation guidelines, and our analysis of data, confirm this view: most hedge scopes coincided with constituent yields in the parse tree. However, as we noted in the previous chapter, there were some hedge instances where that was not the case. The case of attributive adjectives was an example: their scope affected the attributed noun, but excluded other premodifiers and complements of the noun phrase. This seems to contrast with the mentioned semantic definition of scope: it looks like the part of the sentences to which the tentativeness applies should be the whole noun phrase. A similar situation happens with adverbs and nouns: typically, they exclude premodifiers such as determiners or adjectives. Even when, through the learning process, we introduced rules to cope with these cases, the question remains to whether they should not be originally included in the scope.

Another, similar, situation is the case of verbs, where the scope generally matches the syntactic yield of the parent verb phrase of the hedge cue, excluding the sentence subject. However, in the case of passive voice usage, the scope is the whole clause, because in this case the original verb object becomes the sentence subject. If we consider that the use of passive voice constructions does not change the semantic value of a proposition and applying the semantic definition of scope, we wonder whether the scope of verbs should not always be the whole clause. This seems an interesting linguistic research topic, which we leave for future work.

6.5 Future Work

Generalizing the comment made in the last paragraph, we think that it could be very useful to conduct a linguistic study of what exactly the scope of a hedge cue should be, differentiating the distinct part-of-speech of the hedge cues.

Another possible research direction is to better determine when the use for hedge cues actually corresponds to the author's uncertainty about his/her assertions, or when it expresses a pragmatic position, seeking to gain reader acceptance of claims, an aspect thoroughly studied by Hyland. This seems a much more difficult task, probably involving an elaborate semantic and pragmatic analysis of the context surrounding the hedging.

Although an important body of work has been done on the computational identification of uncertain sentences, it is still unclear how useful these results are on more general NLP tasks, such as information extraction or text mining. The use of speculative language detection to identify not-so-certain extracted relations has, to the best of our knowledge, yet to be done. For that task, the performance measure here presented should be re-analyzed: probably, there are errors more relevant (e.g. wrongly including a proposition within a hedge scope) than others (excluding the initial noun phrase determiner from the scope).

With respect to the methodology, it could be useful to evaluate it on different problems, and try to somehow characterize the tasks it could potentially be successfully applied to. The number of learning instances, the structure and the number of learning attributes and the ability of the domain expert to induce, from the learning attributes for the problem, the possible causes for misclassification are potential factors for further study.

Appendix A

List of Scope Classification Errors

This appendix show the full list of scope identification errors on the evaluation corpus. Each row shows the correct and guessed scope, with bold text indicating the part of the sentence wrongly included or wrongly ommited in the guessed scope. The Matching Ancestor and Matching Ancestor (suggested) attribute signals the coincidence of the original and guessed scope with the syntactic scope of some ancestor of the first word in the hedge cue (P corresponds to the parent in the tree, GP to the grandparent, GGP to the great-grandparent and GGGP to the great-great-grandparent). The table also shows the type of each error, according with the categorization given in chapter 5.

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
apparently/RB	apparently due to the imbalance of beta- versus alpha-globin chains	apparently due to the imbalance of beta- versus alpha-globin chains**, leading to the precipitation of excess alpha-globin chains to form Heinz bodies**	NONE	NONE	Parser Error
appear/VB	**impaired activation of NFkappaB does not** appear linked to a reduction of TCRzeta expression	appear linked to a reduction of TCRzeta expression	GGP	P	
appear/VB	appear that components of each P-like element-binding complexes are not identical **and may coordinately contribute to transcriptional activity**	appear that components of each P-like element-binding complexes are not identical	NONE	P	Paser Error
appear/VB	DMDTC does not appear to alter NF-kappaB directly	DMDTC does not appear to alter NF-kappaB directly **as pre-incubation of nuclear extracts with DMDTC does not diminish binding activity of this protein**	NONE	GGP	Parser Error
appeared/VBD	the binding to the E-selectin-NF-kappa B oligomer appeared relatively unaffected even at ratios $>$ 400**, i.e., those achieved in EC treated with 40 mM NAC**	the binding to the E-selectin-NF-kappa B oligomer appeared relatively unaffected even at ratios $>$ 400	GGP	GP	Modified Scope Error
either_or/CC	either not produced due to block of transcription of their respective genes (Oct-2, OBF-1, PU.1), or are rendered inactive posttranslationally (NF-kappa B, E47)	**These transcription factors are** either not produced due to block of transcription of their respective genes (Oct-2, OBF-1, PU.1), or are rendered inactive post-translationally (NF-kappa B, E47)	NONE	GGP	Ancestor Selection Error
either_or/CC	either independent or only partially dependent	either independent or only partially dependent **on MEK1 and MEK2**	P	GP	Other
elusive/JJ	**The mechanism by which progesterone causes localized suppression of the immune response during pregnancy has remained elusive**	No suggested scope	NONE	P	Parser Error
indicate_that/VBP	indicate that Bcl-2 acts downstream of MAPK kinase-1 but upstream of MAPK	indicate that Bcl-2 acts downstream of MAPK kinase-1 but upstream of MAPK **and suggest that, in the signaling pathway of the apoptotic process induced by bufalin, the transcriptional activity of activator protein-1 may be down-regulated through the inhibition of MAPK activity by Bcl-2**	NONE	P	Paser Error

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
indicate_that/VBP	indicate that tissue specificity of gene expression is not accompanied by drastic changes in gene nuclear topography	indicate that tissue specificity of gene expression is not accompanied by drastic changes in gene nuclear topography**, rather suggesting that gene organization within the nucleus may be primarily dependent on structural constraints imposed on the respective chromosomes**	NONE	P	Ancestor Selection Error
indicate_that/VBP	indicate that iNO attenuates iNOS expression in macrophages and inhibits monocyte adhesion to endothelial cells	indicate that iNO attenuates iNOS expression in macrophages and inhibits monocyte adhesion to endothelial cells**, and suggest that endogenously derived iNO may be an important autoregulatory inhibitor of vascular inflammation**	NONE	P	Parser Error
indicate_that/VBP	indicate that the adhesion of RA synovial cells to matrices such as hyaluronic acid through CD44 could up-regulate VCAM-1 expression **and VCAM-1-mediated adhesion to T cells, which might in turn cause activation of T cells and synovial cells in RA synovitis**	These results indicate that the adhesion of RA synovial cells to matrices such as hyaluronic acid through CD44 could up-regulate VCAM-1 expression	NONE	GP	Parser Error
indicated_that/VBD	indicated that there was no significant change in 17beta-HSD IV or DNA-PK (CS) mRNA levels following treatment with 1,25 (OH) 2D3**, although expression of both genes varied with changes in cell proliferation**	indicated that there was no significant change in 17beta-HSD IV or DNA-PK (CS) mRNA levels following treatment with 1,25 (OH) 2D3	NONE	P	Modified Scope Error
indicated_that/VBN	indicated that protein tyrosine phosphatase (PTPase) inhibitors can down-modulate the tumor necrosis factor (TNF)-mediated activation of the nuclear transcription factor NF-kappa B in ML-1a, a monocytic cell line	indicated that protein tyrosine phosphatase (PTPase) inhibitors can down-modulate the tumor necrosis factor (TNF)-mediated activation of the nuclear transcription factor NF-kappa B in ML-1a, a monocytic cell line **(Singh and Aggarwal, J. Biol. Chem. 1995: 270: 10631)**	NONE	P	Other
likely/JJ	RelA homodimers are likely to play a dominant role in TNF-alpha-induced ICAM-1 transcription in monocytic cells	**in vivo** RelA homodimers are likely to play a dominant role in TNF-alpha-induced ICAM-1 transcription in monocytic cells	NONE	GGP	Other

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
likely/RB	**Aggregation was** likely caused by the up-regulated surface expression of adhesion molecules including integrin alpha, L-selectin, ICAM-3, and H-CAM	likely caused by the up-regulated surface expression of adhesion molecules including integrin alpha, L-selectin, ICAM-3, and H-CAM	GGP	NONE	Ancestor Selection Error
may/MD	**This difference in efficacy, which correlates with the compounds' anti-inflammatory potency in vivo,** may be explained by differences in structure and conformation	may be explained by differences in structure and conformation	GP	P	Ancestor Selection Error
may/MD	**Nuclear factor-kappa B (NF-kappa B)/Rel transcription factors** may be involved in atherosclerosis	may be involved in atherosclerosis	GP	P	Ancestor Selection Error
may/MD	**coordination of NF-kappaB and STAT6** may be required for induction of germline Cepsilon transcription by IL-4	may be required for induction of germline Cepsilon transcription by IL-4	GP	P	Ancestor Selection Error
may/MD	may be due to the fact that T cells treated with IL-2 contained those located in S + G2/M phases of the cell cycle**, whereas the vast majority of T cells treated with IFN-alpha/beta were located in G0G1 phase**	may be due to the fact that T cells treated with IL-2 contained those located in S + G2/M phases of the cell cycle	NONE	P	Other
may/MD	may be involved in T cell activation as important negative regulators of the transcription factor AP1	**protein phosphatases 1 and 2A (PP1 and PP2A)** may be involved in T cell activation as important negative regulators of the transcription factor AP1	P	GP	Passive Voice Annotation Error
may/MD	may provide an explanation for the chronicity of the disease	may provide an explanation for the chronicity of the disease**, and may identify a novel therapeutic target in this inflammatory vasculopathy**	P	GP	Ancestor Selection Error
may/MD	may be down-regulated through the inhibition of MAPK activity by Bcl-2	**the transcriptional activity of activator protein-1** may be down-regulated through the inhibition of MAPK activity by Bcl-2	P	GP	Passive Voice Annotation Error
may/MD	may be regulated by distinct mechanisms	**this signaling branch** may be regulated by distinct mechanisms	P	GP	Passive Voice Annotation Error
may/MD	may be accounted for in part by the enhancement of GLUT1 and GLUT4 expression through PPARgamma activation	**the mechanism by which YM268 increased glucose uptake,** may be accounted for in part by the enhancement of GLUT1 and GLUT4 expression through PPARgamma activation	P	GP	Passive Voice Annotation Error

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
may/MD	may be involved in the pathogenesis of inflammation, including RA synovitis	**such cross-talking among distinct adhesion molecules** may be involved in the pathogenesis of inflammation, including RA synovitis	P	GP	Passive Voice Annotation Error
might/MD	**cAMP** might be involved in c-fos induction via H2 receptors	might be involved in c-fos induction via H2 receptors	GP	P	Ancestor Selection Error
might/MD	**cellular genes** might be activated by Ad2 virus infection in nonpermissive cells where no viral gene products could be detected	might be activated by Ad2 virus infection in nonpermissive cells where no viral gene products could be detected	GP	P	Ancestor Selection Error
might/MD	might in turn **cause activation of T cells and synovial cells in RA synovitis**	might in turn	NONE	P	Parser Error
must/MD	must be short-lived	must be short-lived**, apparently due to the imbalance of beta- versus alpha-globin chains, leading to the precipitation of excess alpha-globin chains to form Heinz bodies**	NONE	P	Other
or/CC	by A beta peptides and IFN gamma, or by LPS	**the mechanisms underlying the inducible expression of kappa B-dependent genes in microglia stimulated** by A beta peptides and IFN gamma, or by LPS	NONE	GGP	Parser Error
or/CC	by known inhibitors of NF-kappa B or by simultaneous treatment of the cells **with surfactant lipids**	by known inhibitors of NF-kappa B or by simultaneous treatment of the cells	NONE	P	Parser Error
or/CC	TIMP-2 or a synthetic metalloproteinase inhibitor (BB-94)	**but not** TIMP-2 or a synthetic metalloproteinase inhibitor (BB-94)	NONE	P	Parser Error
or/CC	independent of, or in concert with	independent of, or in concert with**,**	NONE	P	Other
possibility/NN	possibility is that in the absence of gamma (c), the IL-4R alpha chain is able to transduce signals by homodimerization	**An alternative** possibility is that in the absence of gamma (c), the IL-4R alpha chain is able to transduce signals by homodimerization	NONE	GP	Noun Annotation Error
possibility/NN	possibility that Stat3gamma may be transcriptionally inactive and may compete with Stat3alpha for Stat3 binding sites in these terminally differentiated myeloid cells	**our findings suggest the** possibility that Stat3gamma may be transcriptionally inactive and may compete with Stat3alpha for Stat3 binding sites in these terminally differentiated myeloid cells	NONE	GGGP	Noun Annotation Error

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
possible/JJ	possible relationship between the activation of the electron-transport system at the plasma membrane **by ascorbate or its free radical and redox-dependent gene transcription in T-cells**	possible relationship between the activation of the electron-transport system at the plasma membrane	NONE	P	Parser Error
possible/JJ	possible mechanism of cellular transformation **by human T-cell leukemia virus type 1**	one possible mechanism of cellular transformation	NONE	P	Parser Error
possible/JJ	possible targets	**other** possible targets	NONE	P	Adjective Annotation Error
possible/JJ	possible importance	possible importance **such as CREB, CTF, OTF-1, and OTF-2**	NONE	P	Adjective Annotation Error
possible/JJ	possible mechanisms **underlying endothelial cell activation by xenogeneic serum**	possible mechanisms	NONE	P	Parser Error
possibly/RB	possibly through NF-kappa B activation	**xenogeneic serum promotes leukocyte-endothelium interaction** possibly through NF-kappa B activation	NONE	GGP	Adverb Annotation Error
potential/JJ	potential pathways for cytokine-induced periodontal tissue damage, mediated by NF-kappa B1	**several** potential pathways for cytokine-induced periodontal tissue damage, mediated by NF-kappa B1	NONE	P	Adjective Annotation Error
potential/JJ	potential role for IL-7 signaling pathways in transformation by v-Abl	potential role for IL-7 signaling pathways in transformation by v-Abl **while demonstrating that a combination of IL-4 and IL-7 signaling can not substitute for an active v-Abl kinase in transformed pre-B cells**	P	GP	Ancestor Selection Error
potentially/RB	potentially regulating eotaxin gene expression and/or mediating the effects of anti-inflammatory drugs	**the regulatory promoter elements** potentially regulating eotaxin gene expression and/or mediating the effects of anti-inflammatory drugs	NONE	GP	Adverb Annotation Error
potentially/RB	potentially dysfunctional T cells **in patients with cancer**	potentially dysfunctional T cells	NONE	GP	Other

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
potentially/RB	potentially could regulate transcription of specific genes	**Sequence-specific DNA-binding small molecules that can permeate human cells** potentially could regulate transcription of specific genes	NONE	GP	Other
potentially/RB	potentially tissue-damaging	potentially tissue-damaging **granulocytes undergo apoptosis before being cleared by phagocytes in a non-phlogistic manner**	NONE	GP	Adverb Annotation Error
presumably/RB	Viral reactivation from latency and spread from this lymphoid reservoir is presumably required for development of nonlymphoid tumors **like KS**	Viral reactivation from latency and spread from this lymphoid reservoir is presumably required for development of nonlymphoid tumors	GGGP	GGP	Modified Scope Error
probably/RB	probably being mediated by both transcriptional and posttranscriptional mechanisms	**effect** probably being mediated by both transcriptional and posttranscriptional mechanisms	NONE	NONE	Adverb Annotation Error
propose/VBP	propose that the activation of an NFAT kinase by PDTC could be responsible for the rapid shuttling of the NFAT, therefore transiently converting the sustained transactivation of this transcription factor that occurs during lymphocyte activation	propose that the activation of an NFAT kinase by PDTC could be responsible for the rapid shuttling of the NFAT, therefore transiently converting the sustained transactivation of this transcription factor that occurs during lymphocyte activation**, and show that c-Jun NH2-terminal kinase (JNK) can act by directly phosphorylating NFATp**	NONE	P	Parser Error
remains_to_be_elucidated/VBZ	**the molecular basis for Th1- and Th2- specific gene expression** remains to be elucidated	remains to be elucidated	GP	P	Ancestor Selection Error
seem/VBP	**levels of released** TNF-alpha seem to correlate with the stage of B-cell maturation	TNF-alpha seem to correlate with the stage of B-cell maturation	NONE	GP	Parse Error
seems/VBZ	Enhanced IL-5 production by helper T cells seems to cause the eosinophilic inflammation of both atopic and nonatopic asthma	**CONCLUSION:** Enhanced IL-5 production by helper T cells seems to cause the eosinophilic inflammation of both atopic and nonatopic asthma	NONE	GP	Other
seems/VBZ	Activation of NF-kappa B seems to play an important role in the regulation of many proinflammatory cytokine genes	Activation of NF-kappa B seems to play an important role in the regulation of many proinflammatory cytokine genes**, but can not be the only mechanism**	NONE	GGP	Other
should/MD	**cepharanthine** should be further pursued for its chemotherapeutic potential in HIV-1-infected patients	should be further pursued for its chemotherapeutic potential in HIV-1-infected patients	GP	P	Ancestor Selection Error

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
suggest/VBP	suggest that the CD28 response element (CD28RE) does not function independently but works instead in conjunction with the adjacent promoter proximal AP-1-binding site	suggest that the CD28 response element (CD28RE) does not function independently but works instead in conjunction with the adjacent promoter proximal AP-1-binding site **and this hypothesis is confirmed here**	NONE	P	Parser Error
suggest/VBP	suggest that in future clinical gene therapy trials, a combined pharmacologic and genetic strategy **like the one reported here may improve the survival of transduced cells and prolong clinical benefit**	suggest that in future clinical gene therapy trials, a combined pharmacologic and genetic strategy	NONE	P	Modified Scope Error
suggest/VBP	suggest that**, unlike I kappa B alpha, I kappa B beta is constitutively phosphorylated and resynthesized as a hypophosphorylated form**	suggest that	NONE	P	Modified Scope Error
suggest/VBP	suggest that 1) lipid A myristoyl fatty acid**, although it is important for the induction of inflammatory cytokine production by human monocytes, is not necessary for the induction of Mn SOD, 2) endotoxin-mediated induction of Mn SOD and inflammatory cytokines are regulated, at least in part, through different signal transduction pathways, and 3) failure of the mutant endotoxin to induce tumor necrosis factor-alpha production is, at least in part, due to its inability to activate mitogen-activated protein kinase**	suggest that 1) lipid A myristoyl fatty acid	NONE	P	Modified Scope Error
suggest/VBP	suggest that the signaling pathway from Fc receptors leading to expression of different genes important to leukocyte biology, initiates with tyrosine kinases and requires MAPK activation**; but in contrast to other tyrosine kinase receptors, FcR-mediated MAPK activation does not involve Ras and Raf**	suggest that the signaling pathway from Fc receptors leading to expression of different genes important to leukocyte biology, initiates with tyrosine kinases and requires MAPK activation	NONE	P	Other

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
suggest/VBP	suggest that the adherence/contact of SS RBC to endothelial cells in large vessel can generate enhanced oxidant stress leading to increased adhesion and diapedesis of monocytes**, as well as heightened adherence of SS reticulocytes, indicating that injury/activation of endothelium can contribute to vaso-occlusion in SCD**	suggest that the adherence/contact of SS RBC to endothelial cells in large vessel can generate enhanced oxidant stress leading to increased adhesion and diapedesis of monocytes	NONE	P	Modified Scope Error
suggest/VBP	suggest that transcriptional activation of RE/AP is not mediated by NFAT**, because activation of a NFAT reporter is not affected by the addition of CTLA4Ig**	suggest that transcriptional activation of RE/AP is not mediated by NFAT	NONE	P	Modified Scope Error
suggested/VBN	suggested roles of the downstream 3' regions acting as a Locus Control Region (LCR)	**The** suggested roles of the downstream 3' regions acting as a Locus Control Region (LCR)	NONE	P	Adjective Annotation Error
suggesting/VBG	suggesting TNFalpha **and IL-1beta cooperate differently with the cAMP/PKA activation pathway to induce HIV-1 expression in U1 cells**	suggesting TNFalpha	NONE	P	Parser Error
suggesting/VBG	suggesting that induction of GAS-like DNA-protein binding complexes by IL-2, IL-4, and IL-12 is highly selective **and represents one important factor in determining specific gene activation**	suggesting that induction of GAS-like DNA-protein binding complexes by IL-2, IL-4, and IL-12 is highly selective	NONE	P	Parser Error
suggesting/VBG	suggesting a role for HMG-I (Y) in repression **of adult globin genes**	suggesting a role for HMG-I (Y) in repression	NONE	P	Ancestor Selection Error
suggestion/NN	suggestion **that a surrogate gamma'-chain, which can interact with the IL-4R alpha chain to mediate signaling, is expressed on cells lacking gamma (c)**	**the** suggestion	NONE	P	Ancestor Selection Error
suggests/VBZ	suggests a therapeutic approach against AIDS by application of two drugs, one against a cellular and the other a viral target	suggests a therapeutic approach against AIDS by application of two drugs, one against a cellular and the other a viral target**, which may provide an approach to the problem of frequent emergence of resistant variants to combinations of drugs that target only HIV genes**	NONE	P	Other

Hedge Cue	Scope Text	Guessed Scope Text	Matching Ancestor	Matching Ancestor (suggested)	Error Type
suggests/VBZ	suggests a general approach for regulation of gene expression, **as well as a mechanism for the inhibition of viral replication**	suggests a general approach for regulation of gene expression	NONE	P	Modified Scope Error
suspected/VBN	**A role in thymic maturation for factors of the NF-kappaB family has long been suspected**	None	NONE	P	Other
unknown/JJ	unknown function	unknown function **previously reported as an N-Myc interactor**	P	GP	Ancestor Selection Error
whether/IN	**whether and how glucocorticoids affect the function of the inflammatory infiltrate**	None	NONE	P	Other
whether/IN	whether monocyte TF activation occurs in cardiac transplant recipients	**(1)**whether monocyte TF activation occurs in cardiac transplant recipients	NONE	P	Other
whether/IN	whether monocyte TF expression is affected by treatment with cyclosporin A (CsA)	**(2)**whether monocyte TF expression is affected by treatment with cyclosporin A (CsA)	NONE	P	Other
whether/IN	whether xenogeneic serum**, as a source of xenoreactive natural antibodies and complement, induced endothelial cell activation with consequent leukocyte adhesion under flow conditions**	whether xenogeneic serum	NONE	P	Modified Scope Error
whether/IN	whether this fatty acid can modulate endothelial activation**, ie, the concerted expression of gene products involved in leukocyte recruitment and early atherogenesis**	whether this fatty acid can modulate endothelial activation	NONE	P	Modified Scope Error

Table A.1: Errors in scope detection, using gold standard hedge cues.

References

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9 Suppl 11, 2008. ISSN 1471-2105. 3

Ananiadou, S., Kell, D., and Tsuj, J. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579, Dec. 2006. ISSN 01677799. 3

Baker, K., Bloodgood, M., Dorr, B., Filardo, N. W., Levin, L., and Piatko, C. A modality lexicon and its use in automatic tagging. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1402–1407, 2010. 4

Brill, E. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155, Morristown, NJ, USA, 1992. Association for Computational Linguistics. 49

Buchholz, S. and Marsi, E. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. 33, 62, 78

Clyne, M. *The sociocultural dimension: the Dilemma of the German-speaking Scholar*, volume 16 of *Research in text theory*, pages 49–67. de Gruyter, Berlin / New York, 1991. 10

Codd, E. F. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970. ISSN 0001-0782. 62

Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000. ISBN 0521780195. 35

Crompton, P. Hedging in academic writing: Some theoretical problems. *English for Specific Purposes*, 16(4):271–287, 1997. 10

Elkan, C. Log-linear models and conditional random fields. Notes for the course CSE 250B: Principles of Artificial Intelligence: Learning, University of California San Diego, 2013. 77

Falahati, R. A contrastive study of hedging in english and farsi academic discourse. Master's thesis, University of Victoria, 2004. 10

REFERENCES

Farkas, R., Vincze, V., Móra, G., Csirik, J., and Szarvas, G. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 1–12, Uppsala, Sweden, July 2010a. Association for Computational Linguistics. 11

Farkas, R., Vincze, V., Szarvas, G., Móra, G., and Csirik, J., editors. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Uppsala, Sweden, July 2010b. 12, 34, 36, 38, 41, 109, 124

Freund, Y. and Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995. 49

Holmes, J. Doubt and certainty in ESL textbooks. *Applied Linguistics*, 9(1), 1988. 4, 7, 8, 10, 11, 26

Huddleston, R. D. and Pullum, G. K. *The Cambridge Grammar of the English Language*. Cambridge University Press, April 2002. 23, 25

Hyland, K. Hedging in academic writing and EAF textbooks. *English for Specific Purposes*, 13(3): 239–256, 1994. ISSN 08894906. 26

Hyland, K. The author in the text: Hedging scientific writing. *Hongkong Papers in Linguistics and Language Teaching*, 18:33–42, 1995. 4, 5, 6, 8, 10, 11, 26, 81

Hyland, K. Talking to the academy: Forms of hedging in science research articles. *Written Communication*, 13(2):251–281, 1996. 4, 26

Hyland, K. *Hedging in Scientific Research Articles*. Pragmatics & beyond. John Benjamins Publishing Company, 1998. ISBN 9789027250674. 34

Ji, F., Qiu, X., and Huang, X. Detecting hedge cues and their scopes with average perceptron. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 32–39, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 42

Kilicoglu, H. and Bergler, S. Recognizing speculative language in biomedical research articles: A linguistically motivated perspective. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 46–53, Columbus, Ohio, June 2008. Association for Computational Linguistics. 34, 39, 40

Kilicoglu, H. and Bergler, S. A high-precision approach to detecting hedges and their scopes. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 70–77, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 34, 42, 43

Kim, J. D., Ohta, T., Tateisi, Y., and Tsujii, J. Genia corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1, 2003. ISSN 1367-4803. 30, 41

Kim, J.-D. D., Ohta, T., and Tsujii, J. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10+, 2008. ISSN 1471-2105. 30

Klein, D. and Manning, C. D. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics. 69

Kratzer, A. The notional category of Modality. In Portner, P. and Partee, B., editors, *Formal Semantics*, pages 289–323. Blackwell, 1981. 24

Lafferty, J., Mccallum, A., and Pereira, F. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001. 16, 35, 47, 77

Lakoff, G. Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2(4):458–508, October 1973. 4, 25

Li, X., Shen, J., Gao, X., and Wang, X. Exploiting rich features for detecting hedges and their scope. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 78–83, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 36, 37, 42, 43

Liddicoat, A. J. Writing about knowing in science: aspects of hedging in french scientific writing. *LSP and professional communication*, 5(2):8–26, Oct. 2005. 10

Light, M., Qiu, X. Y., and Srinivasan, P. The language of bioscience: Facts, speculations, and statements in between. In Hirschman, L. and Pustejovsky, J., editors, *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17–24, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics. 27, 29, 34, 35, 39, 40, 41

McCallum, A., Freitag, D., and Pereira, F. Maximum entropy markov models for information extraction and segmentation. In *Proceeding 17th International Conference on Machine Learning*, 2000. 47

McCray, A. T., Srinivasan, S., and Browne, A. C. Lexical methods for managing variation in biomedical terminologies. *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pages 235–239, 1994. ISSN 0195-4210. 34

Medlock, B. Exploring hedge identification in biomedical literature. *Journal of Biomedical Informatics*, 41(4):636–654, August 2008a. ISSN 1532-0480. doi: 10.1016/j.jbi.2008.01.001. 35

Medlock, B. *Investigating classification for natural language processing tasks*. PhD thesis, University of Cambridge, June 2008b. 39, 40

Medlock, B. and Briscoe, T. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007. 28, 31, 35, 39, 40

Miller, G. A. WordNet: A lexical database for english. In *Communications of the ACM*, volume 38, pages 39–41, 1995. 34

Mitchell, T. M. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1 edition, Mar. 1997. ISBN 0070428077. 13, 19, 49

Morante, R. personal communication, 2012. 79

Morante, R. and Daelemans, W. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 Workshop*, pages 28–36, Boulder, Colorado, June 2009. Association for Computational Linguistics. 6, 7, 13, 31, 32, 35, 37, 40, 41, 54, 84, 86

Morante, R. and Sporleder, C. Modality and Negation: An introduction to the special issue. *Computational Linguistics*, pages 1–72, Feb. 2012. 24

Morante, R., Liekens, A., and Daelemans, W. Learning the scope of negation in biomedical texts. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 715–724, Morristown, NJ, USA, 2008. Association for Computational Linguistics. 31

Morante, R., Van Asch, V., and Daelemans, W. Memory-based resolution of in-sentence scopes of hedge cues. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 40–47, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 36, 42, 43, 124

Øvrelid, L., Velldal, E., and Oepen, S. Syntactic scope resolution in uncertainty analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 1379–1387, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 34, 43

Özgür, A. and Radev, D. R. Detecting speculations and their scopes in scientific text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, EMNLP '09, pages 1398–1407, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-63-3. 34, 40, 41

Palmer, R. F. *Mood and Modality*. Cambridge Textbooks in Linguistics. Cambridge University Press, New York, 2001. 4, 10, 24

Panocová, R. Expression of modality in biomedical texts. *SKASE Journal of Translation and Interpretation*, pages 81–90, 2008. ISSN 1336-7811. 25

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*, pages 97–104, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 29

Prince, E. F., Frader, J., Bosk, C., and Dipietro, R. J. On hedging in physician-physician discourse. Ablex, 1982. 10

Pyysalo, S., Salakoski, T., Aubin, S., and Nazarenko, A. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7 (Suppl 3):S2+, 2006. ISSN 14712105. 73

Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., and Salakoski, T. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics, special issue*, 9(Suppl 3):S6, 2008. 3

Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989. ISSN 0018-9219. doi: 10.1109/5.18626. 16

Rei, M. and Briscoe, T. Combining manual rules and supervised learning for hedge cue and scope detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 56–63, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 33, 37, 42, 43, 50

Rosá, A. *Identificación de opiniones de diferentes fuentes en textos en español*. PhD thesis, Universidad de la República (Uruguay), Université Paris Ouest (France), Sept. 2011. 56

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. pages 89–114, 1958. 48

Santorini, B. Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report, Department of Computer and Information Science, University of Pennsylvania, 1990. 69

Sauri, R., Verhagen, M., and Pustejovsky, J. Slinket: A partial modal parser for events. In *Language Resources and Evaluation Conference, LREC*, 2006. 24, 27

Settles, B. Biomedical named entity recognition using Conditional Random Fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, JNLPBA '04, pages 104–107, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. 3, 47

Settles, B. Active Learning Literature Survey. Technical report, University of Wisconsin–Madison, 2009. 17, 49

Sha, F. and Pereira, F. Shallow parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 134–141, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. 47, 77, 81

Shatkay, H., Pan, F., Rzhetsky, A., and Wilbur, W. J. Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users. *Bioinformatics*, 24 (18):2086–2093, Sept. 2008. ISSN 1460-2059. 31

Surdeanu, M., Johansson, R., Mayers, A., Marquez, L., and Nivre, J. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 159–177, Manchester, UK, 2008. 33, 62

REFERENCES

Szarvas, G. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *ACL 08: HLT*, 2008. 35, 39, 40

Tang, B., Wang, X., Wang, X., Yuan, B., and Fan, S. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 13–17, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 12, 36, 37, 42

Thompson, P., Venturi, G., McNaught, J., Montemagni, S., and Ananiadou, S. Categorising modality in biomedical texts. In *Proceedings of the LREC 2008 Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 27–34, 2008. 25

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and ichi Tsujii, J. Developing a robust part-of-speech tagger for biomedical text. In Bozanis, P. and Houstis, E. N., editors, *Panhellenic Conference on Informatics*, volume 3746 of *Lecture Notes in Computer Science*, pages 382–392. Springer, 2005. 67, 73

Van Rijsbergen, C. J. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979. 38

Velldal, E., Øvrelid, L., and Oepen, S. Resolving speculation: MaxEnt cue classification and dependency-based scope rules. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 48–55, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 11, 12, 34, 36, 37, 42, 43

Velldal, E., Øvrelid, L., Read, J., and Oepen, S. Speculation and negation: Rules, rankers, and the role of syntax. *Computational Linguistics*, pages 1–64, Feb. 2012. 43, 125, 128

Vincze, V., Szarvas, G., Farkas, R., Mora, G., and Csirik, J. The Bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11):S9+, 2008. ISSN 1471-2105. 6, 8, 11, 27, 29, 30, 41, 65

Vlachos, A. and Craven, M. Detecting speculative language using syntactic dependencies and logistic regression. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–25, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 12

von Fintel, K. *Modality and Language*. MacMillan Reference USA, 2006. 23

Wasserman, L. *All of Statistics: A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, Dec. 2003. ISBN 0387402721. 114

Wilbur, W. J., Rzhetsky, A., and Shatkay, H. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC bioinformatics*, 7(1):356+, July 2006. ISSN 1471-2105. 31

Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005. ISBN 0120884070. 62

Zhang, S., Zhao, H., Zhou, G., and Lu, B.-L. Hedge detection and scope finding by sequence labeling with procedural feature selection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 92–99, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 42

Zhou, H., Li, X., Huang, D., Li, Z., and Yang, Y. Exploiting multi-features to detect hedges and their scope in biomedical texts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 106–113, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 12, 36, 42